Entropy: An Essential Component of Cryptographic Security

Emergent Research Forum

Jeffrey S. Schulman, Jr.
University of Maryland Global Campus
jeffschulman@psu.edu

Executive Summary

Digital computers are *deterministic*, that is, for every input *n,* they will produce output *x*. Cryptographic systems use complex math to encrypt data using *non-deterministically* generated random keys, or *pseudorandom* keys, which are statistically indistinguishable from a random sample. Protecting data and ensuring integrity are core principals of information security. How essential is the use of secure cryptographic ciphers and truly random keys to modern cryptographic security?

Keywords

cryptography, cybersecurity, digital rights management, encryption, entropy, security

Introduction

Information integrity, secrecy, and security are essential for commerce, finance, healthcare, and government computing systems. Contemporary networked computer systems rely heavily on cryptography. Cryptographic security necessitates secret keys which are genuinely random. In this paper, I address the assurance of genuinely frandom key generation, which is a cornerstone principle of modern cryptography. Extinguishment of trust in a system occurs when bad actors acquire the private keys used to secure information. A review of the meaning of cryptography, historical cryptography, cryptographic methods, different types of cryptography, and the absolute importance of secrecy, before entropy, the primary research topic, is explored.

The Committee on National Security Systems Instruction (CNSSI) 4009 (2015) defines cryptography as the "art or science concerning the principles, means, and methods for rendering plain information unintelligible and for restoring encrypted information to intelligible form" (p. 39). Humanity has utilized cryptography for millennia. The first known example of cryptography dates to Augustus Caesar, who was known to use a simple form of encoding which now shares his namesake. Encryption is a robust method to ensure the confidentiality of valuable information or sensitive data. Other methods exist which validate the identity of the sender or recipient of a message and that the contents of a message have not been tampered with or changed.

Identity Management

The composition of an entity's identity are characteristics that uniquely identify that discrete person or thing (CNSSI 4009, 2010). Identity management is essential in ensuring that a data protection scheme is restricted to only authorized people or systems.

Authentication is how an entity proves its identity (Catugono & Galdi, 2014). Authentication can be analog or digital. Analog examples of identity include car titles, birth certificates, and passports. Digital examples include passwords, identification numbers (PINs), and Public Key Infrastructure (PKI) certificates. Authorization and access control comprise the process to challenge a uniquely identifiable entity (also known as a persona), and if that persona can provide a valid identity, allowed access to a restricted area (Catugono & Galdi). One or more factors provide authentication. The composition of multifactor authentication includes something you have, something you know, or something you are. Examples of authentication factors include biometrics (e.g., fingerprints, gait, voiceprint, or retinal scan), passwords, PINs, challenge questions (such as hospital of birth or mother's maiden name), smart cards, identification cards, and radio frequency identification (RFID) tags.

Non-repudiation provides assurance and proof of a sender or recipient sending or receiving a discrete set of information at a specific time. (Vigil, Buchmann, Cabarcas, Weinert, & Wiesmaier, 2015). Non-repudiation is an important factor in legal compliance. Examples of non-repudiation include a person served with a lawsuit, security camera video, or a digitally signed email.

Cryptography

ISO 7498-2:1989 section 3.3.27 defines encryption as "the cryptographic transformation of data to produce ciphertext. Cyphertext is data encoded with a cryptographic key. The conversion of ciphertext back into plaintext requires the use of a decryption key or the exploitation of a flaw in the cryptosystem.

Hashing is a mechanism performing a mathematical function on a data set, and the output of the function is used to validate that the contents of the data set have not changed (CNSSI 4009, 2015). Hashing is a method that ensures data integrity and non-repudiation.

Digital signatures are cryptographic assertions that provide proof of ownership. They utilize a hash function in addition to a PKI signature, which provides for non-repudiation. Digital signatures validate the authenticity of messages (CISSP CBK, 2015).

Cryptographic Ciphers

Cryptographic algorithms (aka ciphers) are mathematic functions. Understanding the history, evolution, and types of ciphers is essential for a complete comprehension of modern cryptosystems.

The original cipher is known as the substitution cipher (a.k.a. Caesar cipher or shift cipher). The shift cipher function shifts letters of plaintext by n characters to produce ciphertext. An example of a shift cipher with a one-letter shift would be the word "dog" becoming "eph." (CISSP CBK, 2015).

A polyalphabetic cipher uses multiple substitution sets which repeat every n characters. For a polyalphabetic cipher with four substitution sets, encipherment would occur over every four letters in the message with the corresponding value of each set. The sets can be either random strings or a shift cipher. An example of a two-set polyalphabetic shift cipher, with the first set having a shift of one and the second set having a shift of two, shows the word "dog" becoming "eqh." (CISSP CBK, 2015).

A one-time pad (a.k.a. Vernam cipher or 'perfect cipher') is a unique cryptosystem through which every ciphertext message has a unique key. For each plaintext message, generate a random string with the same length. This string is used as the key to encrypt the message, shifting each letter in the text by adding the equivalent number (A=1, B=2, C=3, et al.) to the plaintext to create the ciphertext. The pads must never get reused. Consider one-time pads as the only form of perfect encryption, as a full attack against an encoded message will result in every possible combination of the component letters in the message, and no way to determine which one is the actual message (Shannon, 1949).

Stream ciphers apply the cryptographic key and algorithm to each bit in a data stream, one at a time. Block ciphers operate on blocks, or chunks, of data. Divide plaintext into normalized set sizes, and then perform a cryptographic function against each block to produce the ciphertext (CISSP CBK, 2015).

Hashing functions rely on one-way (a.k.a. trapdoor) functions, which are simple to perform in one direction, but reversing the direction requires an immensely significant effort. Hashing functions are utilized to ensure the integrity of data by assuring that there are no changes to the data (Rogaway, & Shrimpton T., 2004).

Cryptographic Methods

Symmetric cryptography utilizes the same key for encryption and decryption. Both parties in a conversation use the same key to convert ciphertext to plaintext and plaintext to ciphertext (CISSP CBK, 2015). Examples of symmetric algorithms include the Triple Data Encryption Algorithm (3DES), the Advanced Encryption Standard (AES), and Blowfish.

Public key (a.k.a. asymmetric) cryptography uses a pair of keys for each conversation. Use one key for encryption and the other for decryption. Symmetric cryptographic

systems rely on one-way functions and are significantly slower than symmetric functions. Asymmetric cryptographic systems often use an asymmetric algorithm for initial handshake, and then utilize symmetric cryptography for performance purposes (Bellare, Desai, Jokipii, & Rogaway, 1997).

Clifford Cocks, a British mathematician working for the United Kingdom government in 1973, developed the first asymmetric algorithm. A similar method was independently re-created by the researchers Ron Rivest, Adi Shamir, and Leonard Adelman in 1977 and named "RSA" after their last names (Rivest, Shamir, & Adleman 1978). RSA is still widely used, and the RSA company has a significant presence in the data security market.

Diffie-Hellman (DH) key exchange is another public key protocol, where two parties who want to communicate without a previously shared secret key can negotiate one over an insecure communications channel. It was published by Whitfield Diffie and Martin Hellman, with initially uncredited assistance by Ralph Merkle, in 1976. DH works in a method where both parties have a private key that stays secret. The parties agree to a random number as a common and publicly shared string. A one-way function is performed, which combines each party's private key with the shared public key and produces a publicly shareable result. Send that result to the other party, who performs a function with their private key, which results in a shared common secret key. Both parties can achieve symmetric cryptography with the common secret (Diffie & Hellman, 1976). Internet applications extensively utilize variations of the DH algorithm for secure web services and virtual private networking.

Data Protection

Protection of data from undesired access or disclosure is known as data loss prevention (DLP). A full data loss prevention scheme covers data in all stages of processing. The three states of data are data at rest (e.g., on a hard disk or memory card), data in use, such as an open file loaded in memory, and data in motion, which is the transfer of data over a network or bus (CNSSI 4009, 2015).

Public key infrastructures (PKIs) are centralized systems comprised of asymmetric key management, issuance, validation, and revocation systems that create and manage signatures associated with public/private key pairs. PKIs act as central authorities that validate the authenticity of computer and user identities. PKI issued certificates are used to secure many different types of digital assets, including websites, email, databases, and executable software (CISSP CBK, 2015).

Webs of trust are decentralized public key exchange systems. Phil Zimmerman created the concept of a web of trust in 1992 with the second version of his message encryption and signature software called Pretty Good Privacy (PGP). In a web of trust, participants aggregate, digitally sign, and share their key store among the members of the network. The aggregate, or a subset, of all the shared keyrings, is checked by a member wanting to exchange data with a recipient. More instances of an individual's public key on trusted keyrings provide a higher level of assurance that the recipient's identity and public key are legitimate. The PGP project is defunct. However, other implementations such as Gnu Privacy Guard (GPG) and OpenPGP are available (Kościelny, Kurkowski, & Srebrny 2013).

Digital Rights Management (DRM) is a system that leverages public-key encryption to restrict the performance of functions against protected data (Nonyelum & Aniche, 2017). Restrictions on content include restricting printing, copying, forwarding, or viewing. Examples of DRM include the Content Scramble System (CSS) for Digital Video Disks (DVDs) and Microsoft Active Directory (AD) Rights Management Service (RMS) for Microsoft Office documents.

Entropy

Hayles (1999) explains that "if information is pattern, then noninformation should be the absence of pattern, that is, randomness" (p. 25). Consider entropy in an information system as a measure of randomness (Shannon, 1948). Guaranteed randomness is essential when generating cryptographic keys. Reversible keys are compromisable, which can cause disclosure of encrypted information.

Random number generators (RNG) are utilized to generate bits of entropy in an information system. They function as a seed key generator, that is, providing a random number as an algorithmic input, which creates a private key as output. There are two types of RNGs utilized in information systems: true random number generators (TRNG) and pseudo-random number generators (PRNG).

TRNGs are discrete hardware components that measure natural sources of entropy. High-security finance, government, and healthcare systems all utilize TRNGs to provide entropy for cryptosystems. Conventional sources of natural entropy include background radio signals, measurement of quantum phenomena which occur in various waves (e.g., sound and light), Johnson-Nyqist (1928) noise (thermal agitation of an electronic circuit) and electromagnetic interference. Hardware random number generators often measure multiple sources of natural entropy. TRNGs are non-deterministic and often packaged alongside hardware security modules (HSM), which provide the ability to generate, store, and use keys for specific algorithms in-hardware (Attridge, 2002). A widely-adopted contemporary example of an HSM platform is the Trusted Platform Module (TPM) 2.0, developed by the not-for-profit Trusted Computing Group (TCG, 2014).

Pseudo-random number generators (PRNG) are software algorithms that are built to provide a statistically random bitstream for an operating system. They operate using the method of measuring samples of various components of a computer system. Most modern operating systems include a PRNG. PRNGs are deterministic, that is, for a specific input seed, they produce the same output every time. Most modern internet-connected operating systems (Linux, Windows, BSD, Android, iOS, OSX, et al.) utilize an entropy pool as a seed source. Variable sampling within the system populates the seed pool. Examples include sources such as user interaction via mouse movements and keyboard strokes, bus timings, hardware drivers, network traffic, or hardware random number generators.

The primary certification program for hardware and software cryptographic modules is the Cryptographic Module Validation Program (CVPM), which is a joint effort of the governments of Canada and the United States. Federal Information Processing Standard (FIPS) 140-2 specifies the security requirements for cryptographic modules. All U.S. and Canadian finance, government, and healthcare information systems that utilize

cryptography must only implement CVMP NIST 140-2 certified cryptomodules. ISO/IEC 19790:2012, "Security requirements for cryptographic modules," and ISO Standard ISO/IEC 24759:2017, "Test requirements for cryptographic modules" are standards that NIST is considering as a reference for an updated standard, NIST SP 140-3 (NIST, 2017).

Cryptosystem Threats

A cryptosystem is only as secure as its weakest component (Tasoluk & Tanrikulu, 2011). There are many ways by which an attacker can potentially compromise encrypted information. The use of a key length, which is known to be easily exhaustible: that is, readily searched for all potential solutions, is a temporal threat. Analysis of cyphertext which uses an algorithm with a flaw, or a known secret key, is a threat. Compromise on an endpoint, such as malware or unauthorized access, can reveal private keys.

Other threats include the cloned VM/snapshot replay threat, whereby an attacker gets a copy of the state of a virtual machine (a computer which runs in a hardware-agnostic portable container) and can reboot the system to disclose the seed for the PRNG (Ristenpart & Yilek, 2010). Nation-states have been caught creating intentional vulnerabilities in systems (Schneier, Fredrikson, Kohno, T. & Ristenpart, 2015). Flaws or trojans in hardware can reduce the entropy in a keyspace, reducing the overall key length and weakening the strength of the cipher (Bhunia, Hsiao, Banga, & Narasimhan, 2014). Entropy starvation exists when an insufficient amount of entropy bits exists in a system's entropy pool (Vassilev & Staples 2016). Entropy starvation creates conditions that allow for key duplication. Heninger, Durumeric, Wustrow, & Halderman (2012) assessed nearly 39 million internet-connected hosts and found that 10% of SSH and 5% of TLS hosts shared private keys because if insufficient entropy and they were able to re-create the private keys of 0.5% of HTTPS hosts and 1% of SSH hosts.

Historical and Contemporary Compromises

The Enigma cipher is a classic example of broken cryptography caused by a lack of entropy. The German government used Enigma machines utilizing the cipher during World War II. The operators of the device were supposed to seed their keyspace with a random set every day, but some operators ignored this policy and reused seeds repeatedly. Coupled with a flaw in the system which didn't allow the same letter to encrypt itself, intercepted ciphertext was vulnerable to cryptanalysis, and the allies succeeded in breaking the cryptography (Gillogly, 1995).

The Content Scramble System (CSS) is a more recent example of a broken cryptosystem. During the development of digital video discs (DVD), a cryptosystem was designed to secure the disks against copying. Due to export controls that were in place, 40-bit encryption was the longest keyspace available. The computing power of a single personal computer available in 1999 was sufficient to brute force the entire keyspace in 18 seconds (Stevenson, 1999).

Every implementation of wireless networking standard 802.11 encryption has vulnerabilities. Wireless Equivalent Privacy (WEP) was the first to be exploited.WEP is a

wireless communication security protocol that utilized a 40 or 104-bit keyspace. Constant cycling of keys creates a keystream, which is meant to function as a Vernam cipher. A weakness in the PRNG used to generate the keystream causes the key scheduling algorithm to be deterministic (Fluhrer, Mantin & Shamir, 2001).

Wifi Protected Access (WPA) was a stop-gap upon disclosure of the weakness of WEP. WPA has numerous vulnerabilities, many of which exploit the use of a non-random pre-shared key (PSK). The key reinstallation attack (KRACK) proved a vulnerability in WPA and the second version of Wifi Protected Access (WPA2). Android and Linux devices were especially susceptible to this attack. The method of exploitation utilized repeated reuse in the nonce (a random number only used once) used for session negotiation. Linux and Android devices were susceptible to an additional defect where the nonce could be forced to zero and to cause immediate disclosure of the private key. A patch is available, which detects this handshake replay and mitigates the vulnerability (Vanhoef & Piessens, 2017).

The 'don't use hardcoded keys' (DHUK) attack leverages a flaw in the outdated, but still widely employed, American National Standards Institute (ANSI) standard x9.31 PRNG. A vulnerability exists in this standard where a hard-coded seed is allowed by design. Many vendors chose to use hard-coded seeds in their implementation of the standard, which reduced entropy and shortened the effective keyspace (Cohney, Green & Heninger, 2017).

Nemec, Sys, Svenda, Klinec, and Matyas (2017) proved a significant loss of entropy in the Infineon RSALibkey generation library. The attack exploits a weakness in the method used to select the prime numbers by which the library generates keys. This flaw affects an estimated quarter of all existing TPM devices and millions of smart cards. The government of Estonia decided to cancel immediately and reissue all national ID cards entirely. This attack is called the *Return of the Coppersmith Attack* (ROCA).

Threat Reduction

Trusted endpoints are necessary as they are by nature allowed to access secret information. Several widely accepted practices exist for cryptosystem hardening. The use of non-standard or irregular ciphers can expose a system to compromise. Select appropriate algorithms and key lengths. Never use irreplaceable keys. Keep systems patched. The National Security Agency and Department of Defense Commercial Solutions for Classified (CSFC) program utilize double encryption, utilizing separate vendors for each layer of encipherment

Conclusion

Trusted computing requires that every endpoint with access to the data has a level of assurance which appropriately protects data from disclosure to unauthorized parties. Cryptographic methods can be used to secure sensitive information systems robustly. Cryptographic keys must not be disclosed to unauthorized parties and must be truly random. Ensuring a dependable source of entropy for key generation is essential.

Multiple sources of entropy can also be used in parallel or simultaneously for random number generation. Another key consideration is to ensure the available entropy in a

system cannot be exhausted. Vassilev and Staples (2016) propose a novel architecture for Internet-connected systems called Entropy-as-a-Service (EaaS), which utilizes entropy and timestamps from a decentralized root of trust. As of writing, commercial products exist which provide network-based sources of entropy (Huges and Nordholt, 2016).

Data-at-rest, data-in-motion, and processing nodes must all be secured and trusted to maintain cryptographic secrecy. Entropy is a necessary and essential component of secure key generation for trusted computing.

REFERENCES

Attridge, J. (2002). *An overview of hardware security modules*. SANS Institute InfoSec Reading Room. Retrieved from https://www.sans.org/reading-room/whitepapers/vpns/overview-hardware-security-modules-757

Bellare, M., Desai, A., Jokipii, E., & Rogaway, P. (1997). Concrete security treatment of symmetric encryption. Annual Symposium On Foundations Of Computer Science - Proceedings, 394-403

Bhunia, S., Hsiao, M. S., Banga, M., & Narasimhan, S. (2014). Hardware Trojan Attacks: Threat Analysis and Countermeasures. *Proceedings of the IEEE*, *102*(8), 1229-1247. doi:10.1109/jproc.2014.2334493

Catuogno, L., & Galdi, C. (2014). Achieving interoperability between federated identity management systems: A case of study. Journal of High Speed Networks, 20(4), 209-221. doi:10.3233/JHS-140499

Cohney, S., Green, M., & Heninger, N. (2017). *Practical state recovery attacks against legacy RNG implementations*. Retrieved from https://duhkattack.com/paper.pdf

Committee on National Security Systems. (2015). Committee on National Security Systems (CNSS) glossary (CNSSI 4009). Retrieved from U.S. Dept. of Commerce, National Institute of Standards and Technology website: https://www.cnss.gov/CNSS/issuances/Instructions.cfm

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, *22*(6), 644-654. doi:10.1109/tit.1976.1055638

Eskicioglu, A. M., & Delp, E. J. (2001). An overview of multimedia content protection in consumer electronics devices. *Signal Processing: Image Communication*, *16*(7), 681-699. doi:10.1016/s0923-5965(00)00050-3

Fluhrer, S., Mantin, I., & Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4. *Selected Areas in Cryptography*, 1-24. doi:10.1007/3-540-45537-x_1

Gillogly, J. J. (1995). Ciphertext-only cryptanalysis of Enigma. *Cryptologia*, *19*(4), 405-413. doi:10.1080/0161-119591884060

Gordon, A., Malik, J., & Hernandez, S. (Eds.). (2015). *Official (ISC)² Guide to the CISSP CBK*

Hayles, N. K. (1999). *How we became posthuman: Virtual bodies in cybernetics, literature and informatics*. Chicago, IL: Univ. of Chicago Press.

Heninger, N., Durumeric, Z., Wustrow, E., & Halderman, J. A. (2012). Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices. In *21st USENIX Security Symposium*. Retrieved from

https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final228.pdf

Huges, R., & Nordholt, J. (2016, January). Strengthening the security foundation of cryptography with Whitewood's quantum-powered entropy engine. Retrieved from https://whitewoodsecurity.com/wp-content/uploads/2017/06/WES_tech_2017_print.pdf

International Organization for Standards (ISO). (1989). Information processing systems - open systems interconnection - basic reference model - part 2: security architecture. (ISO 7498-2:1989).

International Organization for Standards (ISO). (2012). Information technology -- Security techniques -- Security requirements for cryptographic modules (ISO/IEC 19790:2012).

International Organization for Standards (ISO). (2017). Information technology -- Security techniques -- Test requirements for cryptographic modules. (ISO/IEC 24759:2017).

Johnson, J. B. (1928). Thermal Agitation of Electricity in Conductors. *Physical Review*, *32*(1), 97-109. doi:10.1103/physrev.32.97

Kościelny, C., Kurkowski, M., & Srebrny, M. (2016). *Modern Cryptography Primer: Theoretical Foundations and Practical Applications*. Springer-Verlag GmbH.

National Institute of Standards and Technology (NIST). (2002). Federal Information Processing Standard 140-2 (FIPS PUB 140-2)

National Institue of Standards and Technology (NIST). (2017, August). FIPS 140-3 Development | CSRC. Retrieved from https://csrc.nist.gov/Projects/FIPS-140-3-Development

National Security Agency. (2017). Commercial solutions for classified (CSfC) program capability packages. Retrieved from https://www.nsa.gov/resources/everyone/csfc/capability-packages/

Nonyelum, O. F., & Aniche, A. D. (2017). Encryption-based digital right management in internet communication and information transfer. IUP Journal Of Computer Sciences, 11(1), 7-37.

Nyquist, H. (1928). Thermal Agitation of Electric Charge in Conductors. *Physical Review*, *32*(1), 110-113. doi:10.1103/physrev.32.110

Ristenpart, T., Yilek, S. (2010). When good randomness goes bad: virtual machine reset vulnerabilities and hedging deployed cryptography. In: Proceedings of Network and Distributed Security Symposium (NDSS), pp. 1–18. Paper presented at the NDSS Symposium 2010. The Internet Society, San Diego, CA, USA (2010)

Rivest, R., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key components. Communications Of The ACM, 21(2), 120-126.

Rogaway P., & Shrimpton T. (2004) Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: Roy B., Meier W. (eds). Fast Software Encryption. FSE 2004. Lecture Notes in Computer Science, vol 3017. Springer, Berlin, Heidelberg

Schneier, B. (2004, August). Opinion: Cryptanalysis of MD5 and SHA: Time for a new standard. *Computerworld*.

Schneier, B., Fredrikson, M., Kohno, T. & Ristenpart, T. (2015). Surreptitiously weakening cryptographic systems. International Association for Cryptologic

Research (IACR) Cryptology ePrint Archive, 2015:97. Retrieved from https://eprint.iacr.org/2015/097

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3), 379-423. doi:10.1002/j.1538-7305.1948.tb01338.x

Shannon, C. (1949). Communication theory of secrecy systems. Bell System Technical Journal, 28 (4): 656–715, doi:10.1002/j.1538-7305.1949.tb00928.x

Somani, U., Lakhani, K., & Mundra, M. (2010). Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing. 2010 1st International Conference On Parallel Distributed & Grid Computing (PDGC), 211. doi:10.1109/PDGC.2010.5679895

Stevenson, F. (1999). Cryptanalysis of contents scrambling system.

Tasoluk, B., & Tanrikulu, Z. (2011). A Weakest Chain Approach To Assessing the Overall Effectiveness of the 802.11 Wireless Network Security. *International Journal of Wireless & Mobile Networks*, *3*(1), 1-8. doi:10.5121/ijwmn.2011.3101

TPM Library Specification | Trusted Computing Group. (2014, October 1). Retrieved from https://trustedcomputinggroup.org/tpm-library-specification/

Vanhoef, M., & Piessens, F. (2017). Key Reinstallation Attacks. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17*. doi:10.1145/3133956.3134027

Vassilev, A., & Staples, R. (2006). Entropy-as-a-service: unlocking the full potential of cryptography. *Computer*, *49*(9), 98-102. doi:10.1109/MC.2016.275

Vigil, M., Buchmann, J., Cabarcas, D., Weinert, C., & Wiesmaier, A. (2015). Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: A survey. Computers & Security, 5016-32. doi:10.1016/j.cose.2014.12.004