International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 6.392|

||Volume 9, Issue 4, April 2020||

IOT Encryption Algorithms Safety for the IOT Data

Dr. Rajendra Kumar Bharti

Associate Professor, Computer Science & Engineering, Bipin Tripathi Kumaon Institute of Technology, Dwarahat, Uttarakhand, India

ABSTRACT: In cryptography, IoT encryption is the process of encoding information. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext. Ideally, only authorized parties can decipher a ciphertext back to plaintext and access the original information. IoT encryptiondoes not itself prevent interference but denies the intelligible content to a would-be interceptor.

For technical reasons, an IoT encryption scheme usually uses a pseudo-random IoT encryptionkey generated by an algorithm. It is possible to decrypt the message without possessing the key but, for a well-designed IoT encryptionscheme, considerable computational resources and skills are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users.

Historically, various forms of IoT encryption have been used to aid in cryptography. Early IoT encryptiontechniques were often used in military messaging. Since then, new techniques have emerged and become commonplace in all areas of modern computing.[1] Modern IoT encryptionschemes use the concepts of public-key and symmetric-key.[1] Modern IoT encryption techniques ensure security because modern computers are inefficient at cracking the encryption.

KEYWORDS: IoT, encryption, algorithms, data, computers, cryptography, ciphertext, interceptor, scheme

I. INTRODUCTION

One of the earliest forms of IoT encryptionis symbol replacement, which was first found in the tomb of Khnumhotep II, who lived in 1900 BC Egypt. Symbol replacement IoT encryptionis “non-standard,” which means that the symbols require a cipher or key to understand. This type of early IoT encryptionwas used throughout Ancient Greece and Rome for military purposes.[2] One of the most famous military IoT encryptiondevelopments was the Caesar Cipher, which was a system in which a letter in normal text is shifted down a fixed number of positions down the alphabet to get the encoded letter. A message encoded with this type of IoT encryptioncould be decoded with the fixed number on the Caesar Cipher.[3]

Around 800 AD, Arab mathematician Al-Kindi developed the technique of frequency analysis – which was an attempt to systematically crack Caesar ciphers.[2] This technique looked at the frequency of letters in the encrypted message to determine the appropriate shift. This technique was rendered ineffective after the creation of the polyalphabetic cipher by Leon Battista Alberti in 1465, which incorporated different sets of languages. In order for frequency analysis to be useful, the person trying to decrypt the message would need to know which language the sender chose.[2]

Around 1790, Thomas Jefferson theorized a cipher to encode and decode messages in order to provide a more secure way of military correspondence. The cipher, known today as the Wheel Cipher or the Jefferson Disk, although never actually built, was theorized as a spool that could jumble an English message up to 36 characters. The message could be decrypted by plugging in the jumbled message to a receiver with an identical cipher.[4]

A similar device to the Jefferson Disk, the M-94, was developed in 1917 independently by US Army Major Joseph Mauborne. This device was used in U.S. military communications until 1942.[5]

In World War II, the Axis powers used a more advanced version of the M-94 called the Enigma Machine. The Enigma Machine was more complex because unlike the Jefferson Wheel and the M-94, each day the jumble of letters switched to a completely new combination. Each day's combination was only known by the Axis, so many thought the only way to break the code would be to try over 17,000 combinations within 24 hours.[6] The Allies used computing power to severely limit the number of reasonable combinations they needed to check every day, leading to the breaking of the Enigma Machine.

Today, IoT encryption is used in the transfer of communication over the Internet for security and commerce.[1] As computing power continues to increase, computer IoT encryptionis constantly evolving to prevent eavesdropping attacks.[7] With one of the first "modern" cipher suits, DES, utilizing a 56-bit key with 72,057,594,037,927,936 possibilities being able to be cracked in 22 hours and 15 minutes by EFF's DES cracker in 1999, which used a brute-force method of cracking. Modern IoT encryptionstandards often use stronger key sizes often 256, like AES(256-bit mode), TwoFish, ChaCha20-Poly1305, Serpent(configurable up to 512-bit). Cipher suits utilizing a 128-bit or higher key, like AES, will not be able to be brute-forced due to the total amount of keys of 3.4028237e+38 possibilities. The most likely option for cracking ciphers with high key size is to find vulnerabilities in the cipher itself, like inherent biases and backdoors. For example, RC4, a stream cipher was cracked due to inherit biases and vulnerabilities in the cipher. In the context of cryptography, IoT encryptionserves as a mechanism to ensure confidentiality.[1] Since data may be visible on the Internet, sensitive information such as passwords and personal communication may be exposed to potential interceptors.[1] The process of encrypting and decrypting messages involves keys. The two main types of keys in cryptographic systems are symmetric-key and public-key (also known as asymmetric-key).[8][9]

Many complex cryptographic algorithms often use simple modular arithmetic in their implementations.[10]

In symmetric-key schemes,[11] the IoT encryptionand decryption keys are the same. Communicating parties must have the same key in order to achieve secure communication. The German Enigma Machine utilized a new symmetric-key each day for encoding and decoding messages.

In public-key encryption schemes, the IoT encryptionkey is published for anyone to use and encrypt messages. However, only the receiving party has access to the decryption key that enables messages to be read.[12] Public-key IoT encryptionwas first described in a secret document in 1973;[13] beforehand, all IoT encryptionschemes were symmetric-key (also called private-key).[14]:478 Although published subsequently, the work of Diffie and Hellman was published in a journal with a large readership, and the value of the methodology was explicitly described.[15] The method became known as the Diffie-Hellman key exchange.

RSA (Rivest–Shamir–Adleman) is another notable public-key cryptosystem. Created in 1978, it is still used today for applications involving digital signatures.[16] Using number theory, the RSA algorithm selects two prime numbers, which help generate both the IoT encryption and decryption keys.[17]

A publicly available public-key IoT encryption application called Pretty Good Privacy (PGP) was written in 1991 by Phil Zimmermann, and distributed free of charge with source code. PGP was purchased by Symantec in 2010 and is regularly updated.[18]

IoT encryption has long been used by militaries and governments to facilitate secret communication. It is now commonly used in protecting information within many kinds of civilian systems. For example, the Computer Security Institute reported that in 2007, 71% of companies surveyed utilized IoT encryptionfor some of their data in transit, and 53% utilized IoT encryptionfor some of their data in storage.[19] IoT encryptioncan be used to protect data "at rest", such as information stored on computers and storage devices (e.g. USB flash drives). In recent years, there have been numerous reports of confidential data, such as customers' personal records, being exposed through loss or theft of laptops or backup drives; encrypting such files at rest helps protect them if physical security measures fail.[20][21][22] Digital rights management systems, which prevent unauthorized use or reproduction of copyrighted material and protect software against reverse engineering (see also copy protection), is another somewhat different example of using IoT encryptionon data at rest.[23]

IoT encryption is also used to protect data in transit, for example data being transferred via networks (e.g. the Internet, e-commerce), mobile telephones, wireless microphones, wireless intercom systems, Bluetooth devices and bank automatic teller machines. There have been numerous reports of data in transit being intercepted in recent years.[24] Data should also be encrypted when transmitted across networks in order to protect against eavesdropping of network traffic by unauthorized users.[25]

II. DISCUSSION

Conventional methods for permanently deleting data from a storage device involve overwriting the device's whole content with zeros, ones, or other patterns – a process which can take a significant amount of time, depending on the capacity and the type of storage medium. Cryptography offers a way of making the erasure almost instantaneous. This method is called crypto-shredding. An example implementation of this method can be found on iOS devices, where the cryptographic key is kept in a dedicated 'effaceable storage'.[26] Because the key is stored on the same device, this setup on its own does not offer full privacy or security protection if an unauthorized person gains physical access to the device. IoT encryptionis used in the 21st century to protect digital data and information systems. As computing power increased over the years, IoT encryptiontechnology has only become more advanced and secure. However, this advancement in technology has also exposed a potential limitation of today's IoT encryption methods.

The length of the IoT encryption key is an indicator of the strength of the IoT encryptionmethod.[27] For example, the original IoT encryptionkey, DES (Data IoT encryptionStandard), was 56 bits, meaning it had 2^56 combination possibilities. With today's computing power, a 56-bit key is no longer secure, being vulnerable to hacking by brute force attack.[28]

Quantum computing utilizes properties of quantum mechanics in order to process large amounts of data simultaneously. Quantum computing has been found to achieve computing speeds thousands of times faster than today's supercomputers.[29] This computing power presents a challenge to today's IoT encryptiontechnology. For example, RSA IoT encryption utilizes the multiplication of very large prime numbers to create a semiprime number for its public key. Decoding this key without its private key requires this semiprime number to be factored in, which can take a very long time to do with modern computers. It would take a supercomputer anywhere between weeks to months to factor in this key. However, quantum computing can use quantum algorithms to factor this semiprime number in the same amount of time it takes for normal computers to generate it. This would make all data protected by current public-key IoT encryptionvulnerable to quantum computing attacks.[30] Other IoT encryption techniques like elliptic curve cryptography and symmetric key IoT encryptionare also vulnerable to quantum computing.

While quantum computing could be a threat to IoT encryption security in the future, quantum computing as it currently stands is still very limited. Quantum computing currently is not commercially available, cannot handle large amounts of code, and only exists as computational devices, not computers.[31] Furthermore, quantum computing advancements will be able to be utilized in favor of IoT encryption as well. The National Security Agency (NSA) is currently preparing post-quantum IoT encryption standards for the future.[32] Quantum IoT encryptionpromises a level of security that will be able to counter the threat of quantum computing.[31] IoT encryptionis an important tool but is not sufficient alone to ensure the security or privacy of sensitive information throughout its lifetime. Most applications of IoT encryption protect information only at rest or in transit, leaving sensitive data in clear text and potentially vulnerable to improper disclosure during processing, such as by a cloud service for example. Homomorphic encryption and secure multi-party computation are emerging techniques to compute on encrypted data; these techniques are general and Turing complete but incur high computational and/or communication costs.

In response to IoT encryptionof data at rest, cyber-adversaries have developed new types of attacks. These more recent threats to IoT encryptionof data at rest include cryptographic attacks,[33] stolen ciphertext attacks,[34] attacks on IoT encryptionkeys,[35] insider attacks, data corruption or integrity attacks,[36] data destruction attacks, and ransomware attacks. Data fragmentation[37] and active defense[38] data protection technologies attempt to counter some of these attacks, by distributing, moving, or mutating ciphertext so it is more difficult to identify, steal, corrupt, or destroy.[39]

The question of balancing the need for national security with the right to privacy has been debated for years, since IoT encryptionhas become critical in today's digital society. The modern IoT encryption debate[40] started around the '90 when US government tried to ban cryptography because, according to them, it would threaten national security. The debate is polarized around two opposing views. Those who see strong IoT encryption as a problem making it easier for criminals to hide their illegal acts online and others who argue that IoT encryption keep digital communications safe. The debate heated up in 2014, when Big Tech like Apple and Google set IoT encryption by default in their devices. This was the start of a series of controversies that puts governments, companies and internet users at stake.

III. RESULTS

Encryption, by itself, can protect the confidentiality of messages, but other techniques are still needed to protect the integrity and authenticity of a message; for example, verification of a message authentication code (MAC) or a digital signature usually done by a hashing algorithm or a PGP signature. Authenticated encryption algorithms are designed to provide both IoT encryptionand integrity protection together. Standards for cryptographic software and hardware to perform encryption are widely available, but successfully using IoT encryptionto ensure security may be a challenging problem. A single error in system design or execution can allow successful attacks. Sometimes an adversary can obtain unencrypted information without directly undoing the encryption. See for example traffic analysis, TEMPEST, or Trojan horse.[41]

Integrity protection mechanisms such as MACs and digital signatures must be applied to the ciphertext when it is first created, typically on the same device used to compose the message, to protect a message end-to-end along its full transmission path; otherwise, any node between the sender and the IoT encryptionagent could potentially tamper with

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 6.392|

||Volume 9, Issue 4, April 2020||

it. Encrypting at the time of creation is only secure if the IoT encryptiondevice itself has correct keys and has not been tampered with. If an endpoint device has been configured to trust a root certificate that an attacker controls, for example, then the attacker can both inspect and tamper with encrypted data by performing a man-in-the-middle attack anywhere along the message's path. The common practice of TLS interception by network operators represents a controlled and institutionally sanctioned form of such an attack, but countries have also attempted to employ such attacks as a form of control and censorship.[42]

Even when IoT encryptioncorrectly hides a message's content and it cannot be tampered with at rest or in transit, a message's length is a form of metadata that can still leak sensitive information about the message. For example, the well-known CRIME and BREACH attacks against HTTPS were side-channel attacks that relied on information leakage via the length of encrypted content.[43] Traffic analysis is a broad class of techniques that often employs message lengths to infer sensitive implementation about traffic flows by aggregating information about a large number of messages.

Padding a message's payload before encrypting it can help obscure the cleartext's true length, at the cost of increasing the ciphertext's size and introducing or increasing bandwidth overhead. Messages may be padded randomly or deterministically, with each approach having different tradeoffs. Encrypting and padding messages to form padded uniform random blobs or PURBs is a practice guaranteeing that the cipher text leaks no metadata about its cleartext's content, and leaks asymptotically minimal information via its length.[44]

IV. CONCLUSIONS

The export of cryptography is the transfer from one country to another of devices and technology related to cryptography.

In the early days of the Cold War, the United States and its allies developed an elaborate series of export control regulations designed to prevent a wide range of Western technology from falling into the hands of others, particularly the Eastern bloc. All export of technology classed as 'critical' required a license. CoCom was organized to coordinate Western export controls.

Currently, many countries, notably those participating in the Wassenaar Arrangement, have similar restrictions. The Wassenaar restrictions are largely loosensed in the late 2010s

REFERENCES

1. Kessler, Gary (November 17, 2006). "An Overview of Cryptography". Princeton University.
2. ^ "History of Cryptography". Binance Academy. Archived from the original on 2019-04-26. Retrieved 2019-04-02.
3. ^ "Caesar Cipher in Cryptography". GeeksforGeeks. 2016-06-02. Retrieved 2019-04-02.
4. ^ "Wheel Cipher". www.monticello.org. Retrieved 2019-04-02.
5. ^ "M-94". www.cryptomuseum.com. Retrieved 2019-04-02.
6. ^ Hern, Alex (2014-11-14). "How did the Enigma machine work?". The Guardian. ISSN 0261-3077. Retrieved 2019-04-02.
7. ^ Unisys, Dr Glen E. Newton (2013-05-07). "The Evolution of Encryption". Wired. ISSN 1059-1028. Retrieved 2019-04-02.
8. ^ "Key Cryptography – an overview | ScienceDirect Topics". www.sciencedirect.com. Retrieved 2018-02-03.
9. ^ Stubbs, Rob. "Classification of Cryptographic Keys". www.cryptomathic.com. Retrieved 2018-02-03.
10. ^ "Chapter 3. Modular Arithmetic". www.doc.ic.ac.uk. Retrieved 2018-08-15.
11. ^ "Symmetric-key encryption software".
12. ^ Bellare, Mihir. "Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements." Springer Berlin Heidelberg, 2000. p. 1.
13. ^ "Public-Key Encryption – how GCHQ got there first!". gchq.gov.uk. Archived from the original on May 19, 2010.
14. ^ Goldreich, Oded. Foundations of Cryptography: Volume 2, Basic Applications. Vol. 2. Cambridge university press, 2004.
15. ^ Diffie, Whitfield; Hellman, Martin (1976), New directions in cryptography, vol. 22, IEEE transactions on Information Theory, pp. 644–654
16. ^ Kelly, Maria (December 7, 2009). "The RSA Algorithm: A Mathematical History of the Ubiquitous Cryptological Algorithm" (PDF). Swarthmore College Computer Society. Retrieved March 30, 2018.

17. ^ Prasetyo, Deny; Widianto, Eko Didik; Indasari, Ike Pratiwi (2019-09-06). "Short Message Service Encoding Using the Rivest-Shamir-Adleman Algorithm". Jurnal Online Informatika. 4 (1): 39. doi:10.15575/join.v4i1.264. ISSN 2527-9165.
18. ^ Kirk, Jeremy (April 29, 2010). "Symantec buys encryption specialist PGP for $300M". Computerworld.
19. ^ Robert Richardson, 2008 CSI Computer Crime and Security Survey at 19.i.cmpnet.com
20. ^ Keane, J. (13 January 2016). "Why stolen laptops still cause data breaches, and what's being done to stop them". PCWorld. IDG Communications, Inc. Retrieved 8 May 2018.
21. ^ Castricone, D.M. (2 February 2018). "Health Care Group News: $3.5 M OCR Settlement for Five Breaches Affecting Fewer Than 500 Patients Each". The National Law Review. National Law Forum LLC. Retrieved 8 May 2018.
22. ^ Bek, E. (19 May 2016). "Protect Your Company from Theft: Self Encrypting Drives". Western Digital Blog. Western Digital Corporation. Retrieved 8 May 2018.
23. ^ "DRM". Electronic Frontier Foundation.
24. ^ Fiber Optic Networks Vulnerable to Attack, Information Security Magazine, November 15, 2006, Sandra Kay Miller
25. ^ "Data Encryption in Transit Guideline | Information Security Office". security.berkeley.edu.
26. ^ "Welcome". Apple Support.
27. Abood, Omar (July 2018). "A Survey on Cryptography Algorithms". International Journal of Scientific and Research Publications. 6: 495–516 – via ResearchGate.
28. ^ "Encryption methods: An overview". IONOS Digital Guide. Retrieved 2018-10-07.
29. ^ "Quantum computers vastly outperform supercomputers when it comes to energy efficiency". Physics World. 2019-05-01. Retrieved 2018-05-02.
30. ^ Sharma, Moolchand; Choudhary, Vikas; Bhatia, R. S.; Malik, Sahil; Raina, Anshuman; Khandelwal, Harshit (2018-04-03). "Leveraging the power of quantum computing for breaking RSA encryption". Cyber-Physical Systems. 7 (2): 73–92. doi:10.1080/23335777.2019.1811384. ISSN 2333-5777. S2CID 225312133.
31. ^ Solenov, Dmitry; Brieler, Jay; Scherrer, Jeffrey F. (2018). "The Potential of Quantum Computing and Machine Learning to Advance Clinical Research and Change the Practice of Medicine". Missouri Medicine. 115 (5): 463–467. ISSN 0026-6620. PMC 6205278. PMID 30385997.
32. ^ "Post-Quantum Cybersecurity Resources". www.nsa.gov. Archived from the original on 2018-01-18. Retrieved 2018-01-16.
33. ^ Yan Li; Nakul Sanjay Dhotre; Yasuhiro Ohara; Thomas M. Kroeger; Ethan L. Miller; Darrell D. E. Long. "Horus: Fine-Grained Encryption-Based Security for Large-Scale Storage" (PDF). www.ssrc.ucsc.edu. Discussion of encryption weaknesses for petabyte scale datasets.
34. ^ "The Padding Oracle Attack – why crypto is terrifying". Robert Heaton. Retrieved 2016-12-25.
35. ^ "Researchers crack open unusually advanced malware that hid for 5 years". Ars Technica. Retrieved 2016-12-25.
36. ^ "New cloud attack takes full control of virtual machines with little effort". Ars Technica. Retrieved 2016-12-25.
37. ^ Examples of data fragmentation technologies include Tahoe-LAFS and Storj.
38. ^ Burshteyn, Mike (2016-12-22). "What does 'Active Defense' mean?". CryptoMove. Retrieved 2016-12-25.
39. ^ CryptoMove Archived 2018-02-06 at the Wayback Machine is the first technology to continuously move, mutate, and re-encrypt ciphertext as a form of data protection.
40. ^ Catania, Simone. "The Modern Encryption Debate: What's at Stake?". CircleID.
41. ^ "What is a Trojan Virus – Malware Protection – Kaspersky Lab US".
42. ^ Kumar, Mohit (July 2019). "Kazakhstan Begins Intercepting HTTPS Internet Traffic Of All Citizens Forcefully". The Hacker News.
43. ^ Sheffer, Y.; Holz, R.; Saint-Andre, P. (February 2015). Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS) (Report).
44. ^ Nikitin, Kirill; Barman, Ludovic; Lueks, Wouter; Underwood, Matthew; Hubaux, Jean-Pierre; Ford, Bryan (2019). "Reducing Metadata Leakage from Encrypted Files and Communication with PURBs" (PDF). Proceedings on Privacy Enhancing Technologies (PoPETS). 2019 (4): 6–33. arXiv:1806.03160. doi:10.2478/popets-2019-0056. S2CID 47011059.