Deniable encryption

In cryptography and steganography, plausibly **deniable encryption** describes encryption techniques where the existence of an encrypted file or message is deniable in the sense that an adversary cannot prove that the plaintext data exists.[1]

The users may convincingly deny that a given piece of data is encrypted, or that they are able to decrypt a given piece of encrypted data, or that some specific encrypted data exists.[2] Such denials may or may not be genuine. For example, it may be impossible to prove that the data is encrypted without the cooperation of the users. If the data is encrypted, the users genuinely may not be able to decrypt it. Deniable encryption serves to undermine an attacker's confidence either that data is encrypted, or that the person in possession of it can decrypt it and provide the associated plaintext.

In their pivotal 1996 paper, Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky introduced the concept of deniable encryption, a cryptographic breakthrough that ensures privacy even under coercion. This concept allows encrypted communication participants to plausibly deny the true content of their messages. Their work lays the foundational principles of deniable encryption, illustrating its critical role in protecting privacy against forced disclosures. This research has become a cornerstone for future advancements in cryptography, emphasizing the importance of deniable encryption in maintaining communication security.[3] The notion of was used by Julian Assange and Ralf Weinmann in the Rubberhose filesystem.[4][2]

Function

Deniable encryption makes it impossible to prove the origin or existence of the plaintext message without the proper decryption key. This may be done by allowing an encrypted message to be decrypted to different sensible plaintexts, depending on the key used. This allows the sender to have plausible deniability if compelled to give up their encryption key.

Scenario

In some jurisdictions, statutes assume that human operators have access to such things as encryption keys. An example is the United Kingdom's Regulation of Investigatory Powers Act,[5][6] which makes it a crime not to surrender encryption keys on demand from a government official authorized by the act. According to the Home Office, the burden of proof that an accused person is in possession of a key rests on the prosecution; moreover, the act contains a defense for operators who have lost or forgotten a key, and they are not liable if they are judged to have done what they can to recover a key.[5][6]

In cryptography, rubber-hose cryptanalysis is a euphemism for the extraction of cryptographic secrets (e.g. the password to an encrypted file) from a person by coercion or torture[7]—such as beating that person with a rubber hose, hence the name—in contrast to a mathematical or

technical cryptanalytic attack.

An early use of the term was on the sci.crypt newsgroup, in a message posted 16 October 1990 by Marcus J. Ranum, alluding to corporal punishment:

> ...the rubber-hose technique of cryptanalysis. (in which a rubber hose is applied forcefully and frequently to the soles of the feet until the key to the cryptosystem is discovered, a process that can take a surprisingly short time and is quite computationally inexpensive).[8]

Deniable encryption allows the sender of an encrypted message to deny sending that message. This requires a trusted third party. A possible scenario works like this:

1. Bob suspects his wife Alice is engaged in adultery. That being the case, Alice wants to communicate with her secret lover Carl. She creates two keys, one intended to be kept secret, the other intended to be sacrificed. She passes the secret key (or both) to Carl.
2. Alice constructs an innocuous message M1 for Carl (intended to be revealed to Bob in case of discovery) and an incriminating love letter M2 to Carl. She constructs a cipher-text C out of both messages, M1 and M2, and emails it to Carl.
3. Carl uses his key to decrypt M2 (and possibly M1, in order to read the fake message, too).
4. Bob finds out about the email to Carl, becomes suspicious and forces Alice to decrypt the message.
5. Alice uses the sacrificial key and reveals the innocuous message M1 to Bob. Since it is impossible for Bob to know for sure that there might be other messages contained in C, he might assume that there *are* no other messages.

Another scenario involves Alice sending the same ciphertext (some secret instructions) to Bob and Carl, to whom she has handed different keys. Bob and Carl are to receive different instructions and must not be able to read each other's instructions. Bob will receive the message first and then forward it to Carl.

1. Alice constructs the ciphertext out of both messages, M1 and M2, and emails it to Bob.
2. Bob uses his key to decrypt M1 and isn't able to read M2.
3. Bob forwards the ciphertext to Carl.
4. Carl uses his key to decrypt M2 and isn't able to read M1.

Forms of deniable encryption

Normally, ciphertexts decrypt to a single plaintext that is intended to be kept secret. However, one form of deniable encryption allows its users to decrypt the ciphertext to produce a different (innocuous but plausible) plaintext and plausibly claim that it is what they encrypted. The holder of the ciphertext will not be able to differentiate between the true plaintext, and the bogus-claim plaintext. In general, one ciphertext cannot be decrypted to all possible plaintexts unless the key is as large as the plaintext, so it is not practical in most cases for a ciphertext to reveal no information whatsoever about its plaintext.[9] However, some schemes allow

decryption to decoy plaintexts that are close to the original in some metric (such as edit distance).[10]

Modern deniable encryption techniques exploit the fact that without the key, it is infeasible to distinguish between ciphertext from block ciphers and data generated by a cryptographically secure pseudorandom number generator (the cipher's pseudorandom permutation properties).[11]

This is used in combination with some decoy data that the user would plausibly want to keep confidential that will be revealed to the attacker, claiming that this is all there is. This is a form of steganography.

If the user does not supply the correct key for the truly secret data, decrypting it will result in apparently random data, indistinguishable from not having stored any particular data there.

Examples

Layers

One example of deniable encryption is a cryptographic filesystem that employs a concept of abstract "layers", where each layer can be decrypted with a different encryption key. Additionally, special "chaff layers" are filled with random data in order to have plausible deniability of the existence of real layers and their encryption keys. The user can store decoy files on one or more layers while denying the existence of others, claiming that the rest of space is taken up by chaff layers. Physically, these types of filesystems are typically stored in a single directory consisting of equal-length files with filenames that are either randomized (in case they belong to chaff layers), or cryptographic hashes of strings identifying the blocks. The timestamps of these files are always randomized. Examples of this approach include Rubberhose filesystem.

Rubberhose (also known by its development codename Marutukku)[12] is a deniable encryption program which encrypts data on a storage device and hides the encrypted data. The existence of the encrypted data can only be verified using the appropriate cryptographic key. It was created by Julian Assange as a tool for human rights workers who needed to protect sensitive data in the field and was initially released in 1997.[12]

The name Rubberhose is a joking reference to the cypherpunks term rubber-hose cryptanalysis, in which encryption keys are obtained by means of violence.

It was written for Linux kernel 2.2, NetBSD and FreeBSD in 1997–2000 by Julian Assange, Suelette Dreyfus, and Ralf Weinmann. The latest version available, still in alpha stage, is v0.8.3.[13]

Container volumes

Another approach used by some conventional disk encryption software suites is creating a second encrypted volume within a container volume. The container volume is first formatted by filling it with encrypted random data,[14] and then initializing a filesystem on it. The user then fills some of the filesystem with legitimate, but plausible-looking decoy files that the user would

seem to have an incentive to hide. Next, a new encrypted volume (the hidden volume) is allocated within the free space of the container filesystem which will be used for data the user actually wants to hide. Since an adversary cannot differentiate between encrypted data and the random data used to initialize the outer volume, this inner volume is now undetectable. LibreCrypt[15] and BestCrypt can have many hidden volumes in a container; TrueCrypt is limited to one hidden volume.[16]

Other software

- OpenPuff, freeware semi-open-source steganography for MS Windows.
- LibreCrypt, open-source transparent disk encryption for MS Windows and PocketPC PDAs that provides both deniable encryption and plausible deniability.[14][17] Offers an extensive range of encryption options, and doesn't need to be installed before use as long as the user has administrator rights.
- Off-the-Record Messaging, a cryptographic technique providing true deniability for instant messaging.
- StegFS, the current successor to the ideas embodied by the Rubberhose and PhoneBookFS filesystems.
- VeraCrypt (a successor to a discontinued TrueCrypt), an on-the-fly disk encryption software for Windows, Mac and Linux providing limited deniable encryption[18] and to some extent (due to limitations on the number of hidden volumes which can be created[16]) plausible deniability, without needing to be installed before use as long as the user has full administrator rights.
- Vanish, a research prototype implementation of self-destructing data storage.

Detection

The existence of hidden encrypted data may be revealed by flaws in the implementation.[19] It may also be revealed by a so-called watermarking attack if an inappropriate cipher mode is used.[20] The existence of the data may be revealed by it 'leaking' into non-encrypted disk space[21] where it can be detected by forensic tools.[22]

Doubts have been raised about the level of plausible deniability in 'hidden volumes'[23] – the contents of the "outer" container filesystem have to be 'frozen' in its initial state to prevent the user from corrupting the hidden volume (this can be detected from the access and modification timestamps), which could raise suspicion. This problem can be eliminated by instructing the system not to protect the hidden volume, although this could result in lost data.

Drawbacks

Possession of deniable encryption tools could lead attackers to continue torturing a user even after the user has revealed all their keys, because the attackers could not know whether the user had revealed their last key or not. However, knowledge of this fact can disincentivize users from revealing any keys to begin with, since they will never be able to prove to the attacker that they have revealed their last key.[24]

Deniable authentication

Some in-transit encrypted messaging suites, such as Off-the-Record Messaging, offer deniable authentication which gives the participants plausible deniability of their conversations. While deniable authentication is not technically "deniable encryption" in that the encryption of the messages is not denied, its deniability refers to the inability of an adversary to prove that the participants had a conversation or said anything in particular.

This is achieved by the fact that all information necessary to forge messages is appended to the encrypted messages – if an adversary is able to create digitally authentic messages in a conversation (see hash-based message authentication code (HMAC)), they are also able to forge messages in the conversation. This is used in conjunction with perfect forward secrecy to assure that the compromise of encryption keys of individual messages does not compromise additional conversations or messages.

See also

- Chaffing and winnowing – Cryptographic technique
- Deniable authentication – message authentication between a set of participants where the participants themselves can be confident in the authenticity of the messages, but it cannot be proved to a third party after the event
- dm-crypt – Disk encryption software
- Key disclosure law – Legislation that requires individuals to surrender cryptographic keys to law enforcement
- Plausible deniability – Ability to deny responsibility
- Steganography – Hiding messages in other messages
- Unicity distance – Length of ciphertext needed to unambiguously break a cipher

References

1. See http://www.schneier.com/paper-truecrypt-dfs.html Archived (https://web.archive.org/web/20140627184940/https://www.schneier.com/paper-truecrypt-dfs.html) 2014-06-27 at the Wayback Machine. Retrieved on 2013-07-26.
2. Chen, Chen; Chakraborti, Anrin; Sion, Radu (2020). "INFUSE: Invisible plausibly-deniable file system for NAND flash" (https://petsymposium.org/popets/2020/popets-2020-0071.php). *Proceedings on Privacy Enhancing Technologies*. **2020** (4): 239–254. doi:10.2478/popets-2020-0071 (https://doi.org/10.2478%2Fpopets-2020-0071). ISSN 2299-0984 (https://search.worldcat.org/issn/2299-0984). Archived (https://web.archive.org/web/20230208113349/https://petsymposium.org/popets/2020/popets-2020-0071.php) from the original on 2023-02-08. Retrieved 2024-04-02.

3. Ran Canetti, Cynthia Dwork, Moni Naor, Rafail Ostrovsky (1996-05-10). "Deniable Encryption" (http://eprint.iacr.org/1996/002) (PostScript). *Advances in Cryptology – CRYPTO '97*. Lecture Notes in Computer Science. Vol. 1294. pp. 90–104. doi:10.1007/BFb0052229 (https://doi.org/10.1007%2FBFb0052229). ISBN 978-3-540-63384-6. Archived (https://web.archive.org/web/20200824164510/https://eprint.iacr.org/1996/002) from the original on 2020-08-24. Retrieved 2007-01-05.
4. See "Rubberhose cryptographically deniable transparent disk encryption system" (https://web.archive.org/web/20100915130330/http://iq.org/~proff/rubberhose.org/). Archived from the original (http://iq.org/~proff/rubberhose.org/) on 2010-09-15. Retrieved 2010-10-21.. Retrieved on 2009-07-22.
5. "The RIP Act" (https://www.theguardian.com/world/2000/oct/24/qanda). *The Guardian*. London. October 25, 2001. Archived (https://web.archive.org/web/20230328104031/https://www.theguardian.com/world/2000/oct/24/qanda) from the original on March 28, 2023. Retrieved March 19, 2024.
6. "Regulation of Investigatory Powers Bill; in Session 1999-2000, Internet Publications, Other Bills before Parliament" (https://web.archive.org/web/20111108020103/http://www.parliament.the-stationery-office.co.uk/pa/ld199900/ldbills/061/2000061.htm). House of Lords. 9 May 2000. Archived from the original (http://www.parliament.the-stationery-office.co.uk/pa/ld199900/ldbills/061/2000061.htm) on 8 November 2011. Retrieved 5 Jan 2011.
7. Schneier, Bruce (October 27, 2008). "Rubber-Hose Cryptanalysis" (http://www.schneier.com/blog/archives/2008/10/rubber_hose_cry.html). *Schneier on Security*. Archived (https://web.archive.org/web/20090830073523/http://www.schneier.com/blog/archives/2008/10/rubber_hose_cry.html) from the original on August 30, 2009. Retrieved August 29, 2009.
8. Ranum, Marcus J. (October 16, 1990). "Re: Cryptography and the Law..." (https://groups.google.com/group/sci.crypt/msg/86404637e708d900?pli=1) Newsgroup: sci.crypt (news:sci.crypt). Usenet: 1990Oct16.050000.4965@decuac.dec.com (news:1990Oct16.050000.4965@decuac.dec.com). Archived (https://web.archive.org/web/20240402041852/https://groups.google.com/g/sci.crypt/c/W1VUQlC99LM/m/ANkI5zdGQIYJ?pli=1) from the original on April 2, 2024. Retrieved October 11, 2013.
9. Shannon, Claude (1949). "Communication Theory of Secrecy Systems" (https://www.cs.virginia.edu/~evans/greatworks/shannon1949.pdf) (PDF). *Bell System Technical Journal*. **28** (4): 659–664. doi:10.1002/j.1538-7305.1949.tb00928.x (https://doi.org/10.1002%2Fj.1538-7305.1949.tb00928.x). Archived (https://web.archive.org/web/20220114040422/https://www.cs.virginia.edu/~evans/greatworks/shannon1949.pdf) (PDF) from the original on 2022-01-14. Retrieved 2022-01-14.

10. Trachtenberg, Ari (March 2014). *Say it Ain't So - An Implementation of Deniable Encryption* (https://www.blackhat.com/docs/asia-14/materials/Trachtenberg/WP-Asia-14-Trachtenberg-Say-It-Ain%27t-So-An-Implementation-Of-Deniable-Encryption.pdf) (PDF). Blackhat Asia (https://www.blackhat.com/asia-14/). Singapore. Archived (https://web.archive.org/web/20150421042044/https://www.blackhat.com/docs/asia-14/materials/Trachtenberg/WP-Asia-14-Trachtenberg-Say-It-Ain%27t-So-An-Implementation-Of-Deniable-Encryption.pdf) (PDF) from the original on 2015-04-21. Retrieved 2015-03-06.
11. Chakraborty, Debrup; Rodríguez-Henríquez., Francisco (2008). Çetin Kaya Koç (ed.). *Cryptographic Engineering* (https://books.google.com/books?id=nErZY4vYHIoC&q=%22she+should+be+unable+to+distinguish+those+plaintexts%22&pg=PA340). Springer. p. 340. ISBN 9780387718170. Archived (https://web.archive.org/web/20240402041852/https://books.google.com/books?id=nErZY4vYHIoC&q=%22she+should+be+unable+to+distinguish+those+plaintexts%22&pg=PA340#v=snippet&q=%22she%20should%20be%20unable%20to%20distinguish%20those%20plaintexts%22&f=false) from the original on 2024-04-02. Retrieved 2020-11-18.
12. "Rubberhose cryptographically deniable transparent disk encryption system" (https://web.archive.org/web/20120716034441/http://marutukku.org/). *marutukku.org*. Archived from the original (http://marutukku.org/) on 16 July 2012. Retrieved 12 January 2022.
13. "Rubberhose cryptographically deniable transparent disk encryption system" (https://web.archive.org/web/20120716034441/http://marutukku.org/). *marutukku.org*. Archived from the original (http://marutukku.org/) on 16 July 2012. Retrieved 12 January 2022.
14. "LibreCrypt: Transparent on-the-fly disk encryption for Windows. LUKS compatible.: T-d-k/LibreCrypt" (https://github.com/t-d-k/LibreCrypt/blob/master/docs/plausible_deniability.md). *GitHub*. 2019-02-09. Archived (https://web.archive.org/web/20191215031303/https://github.com/t-d-k/LibreCrypt/blob/master/docs/plausible_deniability.md) from the original on 2019-12-15. Retrieved 2015-07-03.
15. "LibreCrypt documentation on Plausible Deniability" (https://github.com/t-d-k/LibreCrypt/blob/master/docs/plausible_deniability.md). *GitHub*. 2019-02-09. Archived (https://web.archive.org/web/20191215031303/https://github.com/t-d-k/LibreCrypt/blob/master/docs/plausible_deniability.md) from the original on 2019-12-15. Retrieved 2015-07-03.
16. "TrueCrypt" (http://www.truecrypt.org/hiddenvolume). Archived (https://archive.today/20120914151319/http://www.truecrypt.org/hiddenvolume) from the original on 2012-09-14. Retrieved 2006-02-16.
17. See its documentation section on "Plausible Deniability" (https://github.com/t-d-k/LibreCrypt/blob/master/docs/plausible_deniability.md) Archived (https://web.archive.org/web/20191215031303/https://github.com/t-d-k/LibreCrypt/blob/master/docs/plausible_deniability.md) 2019-12-15 at the Wayback Machine)
18. "TrueCrypt - Free Open-Source On-The-Fly Disk Encryption Software for Windows Vista/XP, Mac OS X, and Linux - Hidden Volume" (http://www.truecrypt.org/hiddenvolume.php). Archived (https://archive.today/20131015034241/http://www.truecrypt.org/hiddenvolume.php) from the original on 2013-10-15. Retrieved 2006-02-16.

19. Adal Chiriliuc (2003-10-23). "BestCrypt IV generation flaw" (https://web.archive.org/web/20060721044156/http://adal.chiriliuc.com/bc_iv_flaw.php). Archived from the original (http://adal.chiriliuc.com/bc_iv_flaw.php) on 2006-07-21. Retrieved 2006-08-23. {{cite journal}}: Cite journal requires |journal= (help)
20. [title=https://lists.gnu.org/archive/html/qemu-devel/2013-07/msg04229.html Archived (https://web.archive.org/web/20160702002846/https://lists.gnu.org/archive/html/qemu-devel/2013-07/msg04229.html) 2016-07-02 at the Wayback Machine [Qemu-devel] QCOW2 cryptography and secure key handling]
21. "Encrypted hard drives may not be safe: Researchers find that encryption is not all it claims to be" (https://web.archive.org/web/20130330154644/http://news.techworld.com/security/102171/encrypted-hard-drives-may-not-be-safe/). Archived from the original (http://news.techworld.com/security/102171/encrypted-hard-drives-may-not-be-safe/) on 2013-03-30. Retrieved 2011-10-08.
22. http://www.forensicfocus.com/index.php?name=Forums&file=viewtopic&t=3970 Archived (https://web.archive.org/web/20140905190638/http://www.forensicfocus.com/index.php?name=Forums&file=viewtopic&t=3970) 2014-09-05 at the Wayback Machine Is there any way to tell in Encase if there is a hidden truecrypt volume? If so how?
23. "Plausible deniability support for LUKS" (https://gitlab.com/cryptsetup/cryptsetup/wikis/FrequentlyAskedQuestions#c17). Archived (https://web.archive.org/web/20191021003129/https://gitlab.com/cryptsetup/cryptsetup/wikis/FrequentlyAskedQuestions#c17) from the original on 2019-10-21. Retrieved 2015-07-03.
24. "Julian Assange: Physical Coercion" (http://embeddedsw.net/doc/physical_coercion.txt). Archived (https://web.archive.org/web/20130723100802/http://embeddedsw.net/doc/physical_coercion.txt) from the original on 2013-07-23. Retrieved 2011-10-08.

Further reading

- Czeskis, A.; St. Hilaire, D. J.; Koscher, K.; Gribble, S. D.; Kohno, T.; Schneier, B. (2008). "Defeating Encrypted and Deniable File Systems: TrueCrypt v5.1a and the Case of the Tattling OS and Applications" (http://www.usenix.org/event/hotsec08/tech/full_papers/czeskis/czeskis.pdf) (PDF). *3rd Workshop on Hot Topics in Security*. USENIX.
- Howlader, Jaydeep; Basu, Saikat (2009). "Sender-Side Public Key Deniable Encryption Scheme". *Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing*. IEEE. doi:10.1109/ARTCom.2009.107 (https://doi.org/10.1109%2FARTCom.2009.107).
- Howlader, Jaydeep; Nair, Vivek; Basu, Saikat (2011). "Deniable Encryption in Replacement of Untappable Channel to Prevent Coercion". *Proc. Advances in Networks and Communications*. Communications in Computer and Information Science. Vol. 132. Springer. pp. 491–501. doi:10.1007/978-3-642-17878-8_50 (https://doi.org/10.1007%2F978-3-642-17878-8_50).