"Message Authentication Code"

Wikipedia

In cryptography, a **message authentication code** (often **MAC**) is a short piece of information used to authenticate a message.

A MAC algorithm, sometimes called a **keyed** (**cryptographic**) **hash function**, accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC (sometimes known as a *tag*). The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content.

Security

While MAC functions are similar to cryptographic hash functions, they possess different security requirements. To be considered secure, a MAC function must resist existential forgery under chosen-plaintext attacks. This means that even if an attacker has access to an oracle which possesses the secret key and generates MACs for messages of the attacker's choosing, the attacker cannot guess the MAC for other messages without performing infeasible amounts of computation.

MACs differ from digital signatures as MAC values are both generated and verified using the same secret key. This implies that the sender and receiver of a message must agree on the same key before initiating communications, as is the case with symmetric encryption. For the same reason, MACs do not provide the property of non-repudiation offered by signatures specifically in the case of a network-wide shared secret key: any user who can verify a MAC is also capable of generating MACs for other messages. In contrast, a digital signature is generated using the private key of a key pair, which is asymmetric encryption. Since this private key is only accessible to its holder, a digital signature proves that a document was signed by none other than that holder. Thus, digital signatures do offer non-repudiation.

Message integrity codes

The term *message integrity code (MIC)* is frequently substituted for the term MAC, especially in communications,[1] where the acronym MAC traditionally stands for Media Access Control. However, some authors[2] use MIC as a distinctly different term from a MAC; in their usage of the term the MIC operation does not use secret keys. This lack of security means that any MIC intended for use gauging message integrity should be encrypted or otherwise be protected against tampering. MIC algorithms are created such that a given message will always produce the same MIC assuming the same algorithm is used to generate both. Conversely, MAC algorithms are designed to produce matching MACs only if the same message, secret key and initialization vector are input to the same algorithm. MICs do not use secret keys and, when taken on their own, are therefore a much less reliable gauge of message integrity than MACs. Because MACs use secret keys, they do not necessarily need to be encrypted to provide the same level of assurance.

Implementation

MAC algorithms can be constructed from other cryptographic primitives, such as cryptographic hash functions (as in the case of HMAC) or from block cipher algorithms (OMAC, CBC-MAC and PMAC). However many of the fastest MAC algorithms such as UMAC and VMAC are constructed based on universal hashing.[3]

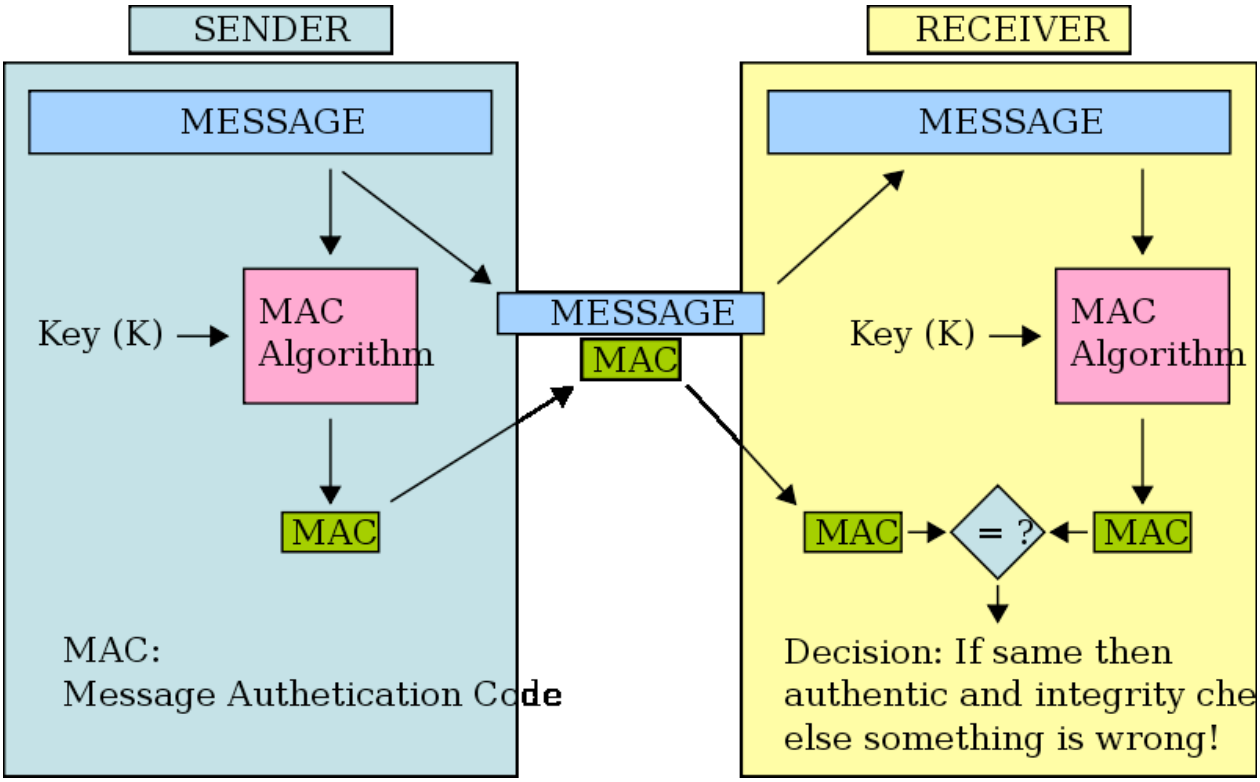
Standards

Various standards exist that define MAC algorithms. These include:

- FIPS PUB 113 *Computer Data Authentication*,[4] withdrawn in 2002,[5] defines an algorithm based on DES.
- ISO/IEC 9797-1 *Mechanisms using a block cipher*[6]
- ISO/IEC 9797-2 *Mechanisms using a dedicated hash-function*[7]

ISO/IEC 9797-1 and -2 define generic models and algorithms that can be used with any block cipher or hash function, and a variety of different parameters. These models and parameters allow more specific algorithms to be defined by nominating the parameters. For example the FIPS PUB 113 algorithm is functionally equivalent to ISO/IEC 9797-1 MAC algorithm 1 with padding method 1 and a block cipher algorithm of DES.

Example



In this example, the sender of a message runs it through a MAC algorithm to produce a MAC data tag. The message and the MAC tag are then sent to the receiver. The receiver in turn runs the message portion of the transmission through the same MAC algorithm using the same key, producing a second MAC data tag. The receiver then compares the first MAC tag received in the transmission to the second generated MAC tag. If they are identical, the receiver can safely assume that the integrity of the message was not compromised, and the message was not altered or tampered with during transmission.

External links

- RSA FAQ's entry on MACs [8]
- Ron Rivest lecture on MACs [9]

References

[1] *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (http://standards.ieee.org/getieee802/download/802.11-2007.pdf). (2007 revision). IEEE-SA. 12 June 2007. doi:10.1109/IEEESTD.2007.373646. .

[2] Fred B Schneider, Hashes and Message Digests, Cornell University (http://www.cs.cornell.edu/courses/cs513/2005fa/NL20.hashing.html)

[3] "VMAC: Message Authentication Code using Universal Hashing" (http://www.fastcrypto.org/vmac/draft-krovetz-vmac-01.txt). *CFRG Working Group* (CFRG Working Group). . Retrieved 16 March 2010.

[4] FIPS PUB 113 *Computer Data Authentication* (http://www.itl.nist.gov/fipspubs/fip113.htm)

[5] Federal Information Processing Standards Publications, Withdrawn FIPS Listed by Number (http://www.itl.nist.gov/fipspubs/withdraw.htm)

[6] ISO/IEC 9797-1 *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher* (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=30656)

[7] ISO/IEC 9797-2 *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function* (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=31136)

[8] http://www.rsasecurity.com/rsalabs/node.asp?id=2177

[9] http://web.mit.edu/6.857/OldStuff/Fall97/lectures/lecture3.pdf

Article Sources and Contributors

Message authentication code *Source*: http://en.wikipedia.org/w/index.php?oldid=412172726 *Contributors*: Abatakar, Ablewisuk, AdjustShift, Andris, Benizi, Bobo192, Ceyockey, Ciphergoth, Connelly, Conny, CryptoDerk, Dachshund, Darkfight, David spector, Davidgothberg, Dimawik, DividedByNegativeZero, Dougher, Ed Brey, Edipo, Emufarmers, Gareth Jones, Giftlite, HamburgerRadio, Henning Makholm, Iridescent, Jesse Viviano, Jorge Stolfi, KnightRider, Kotepho, Linas, Loadmaster, Macabu, Mandarax, Manscher, Matb, Matt Crypto, Maurice Carbonaro, Messiah7, Michael Hardy, Mindmatrix, Mitch Ames, Nealmcb, Nroets, PabloCastellano, Pi.C.Noizecehx, Quarl, Qutezuce, Rasmus Faber, Robertgreer, RonaldDuncan, Schnolle, Sh00tr, Shiningfm, Signalhead, Smilerpt, TonyW, TreasuryTag, Twisp, Varuna, Verloc, Wonderstruck, Ww, Xvani, Y(J)S, 53 anonymous edits

Image Sources, Licenses and Contributors

Image:MAC.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:MAC.svg *License*: Public Domain *Contributors*: User:Twisp

License