Confidentiality Using Authentication

by Robert Schlaff

Within the next ten years, the use of the Internet in both communication and commerce will greatly increase. As a result, the U.S. Government has begun to regulate these activities online. The government wants to regulate communication by decreasing the number of confidential (secret) messages sent so that the government can eavesdrop on illegal activities to prevent crime. Therefore, it has passed regulations on the use (mainly the export) of encryption, the most common method of creating confidentiality. But the government has not regulated "authentication", and here's why. When regulating online banking, the key is to set up ways to verify people's identities on the net. This is most commonly done using authentication. Wider use of authentication spurs the online economy and thus helps U.S. companies that do business online. Therefore, the U.S. government has encouraged use of authentication by leaving it free of all export restrictions and regulations. In this article I hope to show that parties can obtain confidentiality without using encryption.

It is important to keep in mind the differences between confidentiality, encryption, and authentication.

- **Confidentiality** is the passing of information between two parties without a third party being able to understand it.
- **Encryption** is the main way of achieving confidentiality using computers. Encryption transforms a message into a ciphertext using an encryption key. The ciphertext can only be transformed back into the original message using the encryption key and therefore can only be read by someone who has the key. The person who encrypts the message (the sender) and the person who decrypts the message (the receiver) may use the same keys (private key encryption) or different keys (public key encryption). DES, RSA, and IDEA are examples of encryption schemes.
- **Authentication** is the process of verifying that a message was sent by a given person. A message is authenticated by computing a **Message Authentication Code (MAC)**, which is a function of a secret key and the message. This MAC is then appended to the message. A MAC is similar to private key encryption because the sender and receiver both share the same secret key. When using encryption, the secret key allows the receiver to read the message. However, when using authentication, the receiver uses the secret key to decide if the message was sent by the person who claims to have sent it (the only other person with the secret key). If the message was changed or the wrong secret key was used, the MAC will be wrong. This tells the receiver that the message is not authentic. A good authentication algorithm is HMAC-SHA1 [1].

While encryption is the most common way to achieve confidentiality, authentication can also be used. One technique for providing confidentiality using authentication is called ``chaffing and winnowing'' and was first implemented by Ronald Rivest of MIT's Laboratory for Computer Science [2]. The scheme

works by sending **chaff** (incorrect messages) that can only be differentiated from the real message by the intended receiver (who winnows the stream). The word **winnow** means to separate or eliminate useless parts and is derived from its common meaning, to separate wheat from chaff (useless plant parts) on a farm. We will refer to good packets of information as ``wheat'' and bad packets of information as ``chaff''.

Chaffing and winnowing is relatively easy to implement. The sender authenticates the real message by adding a correct MAC to it. Then the sender adds chaff (with incorrect MACs) to confuse any eavesdroppers. The recipient knows which MAC is correct (as he also has the secret key) and throws away all messages with bad MACs (i.e., the chaff) to obtain the original message.

In a simple example of this, Alice sends a message to Bob. Before she sends her message, she authenticates it by computing a MAC and adding it to the end of the message. A secret authentication key can be agreed upon at the beginning by using a method such as Diffie-Hellman. This can be represented by:

message --> message, MAC

Then Alice sends many other messages to Bob with incorrect MACs. It is virtually impossible for an eavesdropper to figure out which is the real message because she does not have access to the secret key. Bob can determine the correct message by recomputing the function of the message and the secret key and comparing it with the MACs appended to the messages. Bob throws outs all messages whose MACs are different from the value he computes.

To make the method work better, the message may be split into packets, each of which may be prefixed with a serial number. This helps us figure out if any packets are lost and how the packets should be ordered. An example of a message before chaff is added would be:

(1, Hi Mike, 34234)
(2, I'll meet you at the post office, 36737)
(3, Kate, 09760)

Adding chaff we get:

(1, Hi Mike, 34234)
(1, Hi Harold, 43243)
(2, I'll meet you at the post office, 36737)
(2, Never speak to me again, 636209)
(3, Kate, 09760)
(3, Sally, 87402)

To obtain the message, the receiver merely discards all the bad packets, retaining the good ones.

What level of confidentiality can be obtained by messages sent by chaffing? The confidentiality of the message is based completely on an eavesdropper's difficulty of distinguishing wheat from chaff. This depends on the MAC algorithm, on how the message is broken into packets, and how the chaffing is done.

A good MAC algorithm (like HMAC-SHA1) will appear to act like a ``random function'' to the eavesdropper. In such cases the adversary will be unable to differentiate between wheat and chaff.

If the adversary sees only one packet with a given serial number, then that packet is probably wheat, not chaff. So a good chaffing process will add at least one chaff packet for each serial number used in the message.

The adversary may also distinguish wheat from chaff by the contents of each packet. If the wheat packets contain English sentences and the chaff packets have random bits, then the adversary will have no difficulty in winnowing wheat from chaff. However, if wheat packets contain a single bit and there are chaff packets with the same serial numbers but complementary bits, the adversary will find it virtually impossible to find the wheat packet. To obtain all the wheat bits and recover the original message, the adversary would have to break the MAC algorithm or know the secret authentication key. With a good MAC algorithm, the adversary's ability to winnow is extremely small, so chaffing provides an excellent degree of confidentiality.

The problem with sending one bit packets is that it creates inefficiency. Assuming we have 32-bit serial numbers and 64-bit MACs, we have to send around 100 bits to transmit one bit of information.

It is possible to make chaffing and winnowing much more efficient, allowing many bits per packet instead of just one. Rivest presents one approach in his paper, using an "all-or-nothing" or "package" transform [3]. This is a keyless (non-encryption) transform that takes the message and produces a ``packaged message'' with the property that the recipient cannot read any part of the message until he has received the entire packaged message. Eve can also read the message once she has eavesdropped on the entire message. All-or-nothing transforms make chaffing and winnowing more efficient because the receiver must know exactly which packets are part of the message in order to read the message.

The sender then breaks the message into 1024-bit chunks, authenticates each block with a MAC, and transmits the result to the receiver. This message is packaged and authenticated, but not encrypted; any eavesdropper could easily reconstruct the message given all of the blocks. However, if a sufficient number of chaff blocks are sent, excellent confidentiality results. For an adversary, the difficulty of determining the original plaintext (obtaining the wheat) is proportional to the number of ways the sub-sequence of the blocks can be chosen and tested for being wheat; this will be exponential in the number of blocks, assuming that the fraction of chaff blocks is guaranteed not to be close to one. When packaging is used, it is not necessary to have as many chaff packets as wheat packets since the adversary must identify the wheat packets precisely (with no omissions or extra packets) in order to retrieve the

message. Thus, for long messages, the relative number of chaff packets can be quite small.

Adding chaff is also very easy: just create a packet with the same serial number in it as the wheat packet, and add a random MAC. The chance that the MAC will be correct is about 1 in 2^64 or about 10^18.

We have demonstrated that excellent confidentiality can be achieved without using encryption, simply by relying on authentication. It would appear that this means that a program that can send confidential messages could be exported. However, export laws have intent clauses in them, so if a company intentionally tries to evade the export laws on encryption, it is guilty of breaking them.

However, chaffing and winnowing makes it difficult for the government to determine intent. Consider the case of Alice, who is sending two messages to Bob with different keys, using an all-in-one transform on both messages. Alice sends packets randomly from message 1 and message 2 to confuse eavesdroppers. To decode message 1, Bob has to collect all of the pieces of message 1 and none of the pieces of message 2 and perform the all-in-one transform on them. Because Bob knows the secret keys, he can separate the two messages and decode them. Eavesdroppers, on the other hand, do not know the secret keys, so they cannot decode either message. It would be very hard for the government to pass regulations to stop Alice from doing this. She is not encrypting anything and she could plausibly say that she did not even mean to send the messages in a confidential manner.

The government used to think that all products without certain encryption routines were unable to produce confidential messages. However, we have established that chaffing and winnowing makes it possible to use authentication to establish confidentiality. In light of this development, it will be much harder for the U.S. government to regulate confidentiality in the future.

References

1

Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-hashing for Message Authentication," RCF2104, February 1997. (Available at ftp://ds.internic.net/rfc/rfc2104.txt)

2

Rivest, R. "Confidentiality without encryption," March 18, 1998. (Available at http://theory.lcs.mit.edu/~rivest/chaffing.txt)

3

Rivest, R. "All-Or-Nothing Encryption and the Package transform," Proceedings of the 1997 Fast Software Encryption Conference (Springer, 1997). (Also available on http://theory.lcs.mit.edu/~rivest/fusion.ps)

Robert Schlaff is a Senior at Yale University. He is the Chairman of the newly formed Chapter of the

Yale Student ACM Chapter and *The Yale Record*, the nation's oldest college humor magazine. In his spare time he enjoys skiing, swimming, and eating small, frequent, and well-spaced meals.