

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

A major goal of security techniques is ``confidentiality''---ensuring that adversaries gain no intelligence from a transmitted message. There are two major techniques for achieving confidentiality:

- Steganography: the art of hiding a secret message within a larger one in such a way that the adversary can not discern the presence or contents of the hidden message. For example, a message might be hidden within a picture by changing the low-order pixel bits to be the message bits. (See Wayner (1996) for more information on steganography.)
- Encryption: transforming the message to a ciphertext such that an adversary who overhears the ciphertext can not determine the message sent. The legitimate receiver possesses a secret decryption key that allows him to reverse the encryption transformation and retrieve the message. The sender may have used the same key to encrypt the message (with symmetric encryption schemes) or used a different, but related key (with public-key schemes). DES and RSA are familiar examples of encryption schemes.

This paper introduces a new technique, which we call ``chaffing and winnowing''---to winnow is to ``separate out or eliminate (the poor or useless parts),'' (Webster's Dictionary), and is often used when referring to the process of separating grain from chaff.

Novel techniques for confidentiality are interesting in part because of the current debate about cryptographic policy as to whether law enforcement should be given when authorized surreptitious access to the plaintext of encrypted messages. The usual technique proposed for such access is ``key recovery,'' where law enforcement has a ``back door'' that enables them to recover the decryption key.

Winnowing does not employ encryption, and so does not have a ``decryption key.'' Thus, the usual arguments in favor of ``key recovery'' don't apply very well for winnowing. As usual, the policy debate about regulating technology ends up being obsoleted by technological innovations. Trying to regulate confidentiality by regulating encryption closes one door and leaves two open (steganography and winnowing).

We now explain how a confidentiality system based on winnowing works. There are two parts to sending a message: authenticating (adding MACs), and adding chaff. The recipient removes the chaff to obtain the original message.

The sender breaks the message into packets, and authenticates each packet using a secret authentication key. That is, the sender appends to each packet a ``message authentication code'' or ``MAC'' computed as a function of the packet contents and the secret authentication key, using some standard MAC algorithm, such as HMAC-SHA1 (see Krawczyk et al. (1997)). We have the transformation of appending a MAC thus:

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

packet --> packet, MAC

The packet is still ``in the clear''; no encryption has been performed. We note that software that merely authenticates messages by adding MACs is automatically approved for export, as it is deemed not to encrypt.

There is a secret key shared by the sender and the receiver to authenticate the origin and contents of each packet---the legitimate receiver, knowing the secret authentication key, can determine that a packet is authentic by recomputing the MAC and comparing it to the received MAC. If the comparison fails, the packet and its MAC are automatically discarded. The sender and the receiver can initially create and agree upon the secret authentication key with any standard technique, such as authenticated Diffie-Hellman.

We note that it is typical for each packet to contain a serial number as well. For example, when a long file is transmitted it is broken up into smaller packets, and each packet carries a unique serial number. The serial numbers help the receiver to remove duplicate packets, identify missing packets, and to correctly order the received packets when reassembling the file. The MAC for a packet is computed as a function of the serial number of the packet as well as of the packet contents and the secret authentication key. As an example, we might have a sequence of the form:

```
(1,Hi Bob,465231)
(2,Meet me at,782290)
(3,7PM,344287)
(4,Love-Alice,312265)
```

of triples of sequence number, message, and MAC.

The second process involved in sending a message is ``adding chaff'': adding fake packets with bogus MACs. The chaff packets have the correct overall format, have reasonable serial numbers and reasonable message contents, but have MACs that are not valid. The chaff packets may be randomly intermingled with the good (wheat) packets to form the transmitted packet sequence. Extending the preceding example, chaff packets might make the received sequence look like:

```
(1,Hi Larry,532105)
(1,Hi Bob,465231)
(2,Meet me at,782290)
(2,I'll call you at,793122)
(3,6PM,891231)
(3,7PM,344287)
(4,Yours-Susan,553419)
(4,Love-Alice,312265)
```

In this case, for each serial number, one packet is good (wheat) and one is bad (chaff). Instead of randomly intermingling the chaff with the wheat, the packets can also be output in sorted order, sorting first by serial number, and then by message contents.

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

To obtain the correct message, the receiver merely discards all of the chaff packets, and retains the wheat packets. But this is what the receiver does anyway! In a typical packet-based communication system the receiver will automatically discard all packets with bad MACs. So the ``winnowing'' process is a normal part of such a system. (Receiving a packet with a bad MAC could conceivably trigger more of a response from the receiver, but not normally; the detection of a missing packet is determined at a different level of the protocol stack, rather than upon receipt of a bad packet, since the packet may have been transmitted more than once and been received OK already.)

Let us verb a word, and let ``chaffing'' mean the process of adding chaff to a sequence of packets. As above, ``winnowing'' is the (usual) process of discarding all packets with bad MACs. We call the good packets ``wheat'' for consistency of metaphor.

How much confidentiality does chaffing provide? This depends on the MAC algorithm, on how the original message is broken into packets, and on how the chaffing is done.

A typical MAC algorithm (such as HMAC-SHA1) will appear to act like a ``random function'' to the adversary, and in such a case the adversary will not be able to distinguish wheat from chaff. It is possible in principle, however, to have an unfortunate MAC algorithm that ``leaks'' information about the message being MAC'ed, allowing the adversary to gain an advantage in distinguishing wheat from chaff. For example, one could define a LEAKY-HMAC-SHA1 MAC algorithm to have an output that is the concatenation of the output of the HMAC-SHA1 algorithm together with the low-order bit of the message being MAC'ed. However, in practice (and in theory) one looks for MAC algorithms that are indistinguishable from random functions, and such algorithms also work fine in a chaffing and winnowing application.

Note that the problem of providing confidentiality by chaffing and winnowing is based on the difficulty (for the adversary) of distinguishing the chaff from the wheat. It is *not* based on the difficulty of breaking an encryption scheme, since there is no encryption being performed (although confidentiality may be obtained nonetheless, just as for steganography).

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

The adversary may also distinguish wheat from chaff by the contents of each packet. If the wheat packets each contains an English sentence, while the chaff packets contain random bits, then the adversary will have no difficulty in winnowing the wheat from the chaff himself.

On the other hand, if each wheat packet contains a single bit, and there is a chaff packet with the same serial number containing the complementary bit, then the adversary will have a very difficult

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

(essentially impossible) task. Being able to distinguish wheat from chaff would require him to break the MAC algorithm and/or know the secret authentication key used to compute the MACs. With a good MAC algorithm, the adversary's ability to winnow is nonexistent, and the chaffing process provides perfect confidentiality of the message contents. To make this clearer with an example, note that the adversary will see triples of the form:

```
(1,0,351216)
(1,1,895634)
(2,0,452412)
(2,1,534981)
(3,0,639723)
(3,1,905344)
(4,0,321329)
(4,1,978823)
...
```

and so on.

I stress that the sending process for chaffing and winnowing is not encryption; it is authentication (adding MACs) followed by adding chaff.

Let us assume that the original message is broken into very short (one-bit) packets, and that MACs have been added to each such packet to create the wheat packets. (There is some obvious inefficiency here, since each wheat packet may end up being, say about 100 bits long, but only transmits one bit. Here each MAC might be 64 bits in length, and each serial number 32 bits long. Additional bits might also be present to identify sender, receiver, etc.)

Such a message sequence is not encrypted, and the process for creating such a message sequence would presumably not be export-controlled, since the message bits are ``in the clear'' and nicely labelled with serial numbers.

The process of creating chaff is also easy: just create a chaff packet with whatever serial number and packet contents you may like, and include a random 64-bit MAC value. This MAC value is overwhelmingly likely to be bad, and thus the packet created is overwhelmingly likely to be chaff. (The chances of creating a good packet are one in 2^{64} ---approximately one in 10^{19} ---which is effectively negligible.) The person creating the chaff (the ``chaffer'') would do so having seen the wheat packets, and would make chaff packets up that have the same serial numbers as the wheat packets do, but with complementary packet contents. Again, it is assumed here that an adversary, not knowing the secret authentication key, can not distinguish a good (wheat) packet from a bad (chaff) one.

It is especially intriguing to now observe that creating chaff does not require knowledge of the secret authentication key! That is, creating chaff is done by creating bogus packets with bogus randomly guessed (and thus bad) MACs; to randomly guess a MAC requires no knowledge of the secret authentication key.

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

We could thus have the following intriguing scenario: Alice is communicating with Bob using a standard packet-based communication scheme. Each packet is authenticated with a MAC created using a secret authentication key known only to Alice and Bob. (In practice, they might use a different key for packets in each direction, although this is not necessary if the packet contents identify sender and receiver.) Furthermore, each packet happens to contain only a single ``message bit.'' (Alice wrote their software, and it contained a bug that caused this unusual behavior.)

So far, Alice and Bob are not encrypting anything, and are using standard messaging techniques that would not be considered as encryption and that would not be export-controlled. Alice and Bob have no intention of achieving confidentiality of their messages from an eavesdropper.

Now, Alice's packets to Bob may be routed from her computer through the computer of her Internet service provider, run by Charles, on another floor of her building, before being sent on to more major trunks of the Internet and then on to Bob.

Charles' computer, for whatever reason, then adds ``chaff'' packets to the packet sequence from Alice to Bob. All of sudden, Charles' activities provide a very high degree of confidentiality for the communications between Alice and Bob! Alice's and Bob's software have not been modified in the least to achieve this confidentiality! Charles does not know the secret authentication key used between Alice and Bob! Alice and Bob did not even want or care to have confidential communications! Charles is not using encryption and does not know any encryption key! Amazing!

Clearly, the cause of the confidentiality is Charles's activities, but Charles has no encryption key or decryption key that he could give to law enforcement. Alice and Bob share an authentication key, but do not perform any encryption, and have no encryption or decryption keys.

Law enforcement may be able to tap the (unencrypted) line from Alice to Charles, but that might be difficult to arrange without Alice's knowledge, as Alice and Charles are in the same building, and may even be friendly or colluding. While Charles' chaffing activities may be suspicious, they don't constitute encryption and don't involve any knowledge of keys on his part; there is no key information he could give to any law enforcement agency.

In a variation on the above scenario, Charles is not ``adding chaff'' but merely multiplexing the stream of packets from Alice to Bob with another stream of packets (say from David to Elaine). To Bob, the stream of packets from David to Elaine looks like chaff, and is discarded. But to Elaine, the converse holds, and she discards the stream of packets from Alice to Bob as chaff. What is wheat to one pair of communicants is chaff to the other pair, and vice versa. Such a situation could arise where Charles is managing a broadcast channel such as a satellite link; here both parties naturally receive the

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

stream of intermingled packets. If the only way to distinguish one stream from another is by the correctness of the MACs, then an adversary will have a hard time separating the streams. (Of course, if there are exactly two streams being multiplexed, then Alice and Bob can read the stream from David to Elaine, and vice versa.)

In such a scenario, the obvious tack for law enforcement to take would be to demand to have access to the secret authentication key shared by Alice and Bob. But access to authentication keys is one thing that government has long agreed that they don't want to have. Having such access would allow the government to forge authentic-looking packets for any pair of parties that are communicating. This is way beyond mere access to encrypted communications, as loss of such authentication keys could wreak massive havoc to the structure and integrity of the entire Internet, allow hackers not only to overhear private messages, but to actually control computers, perhaps to shut down power systems or to airline traffic control systems, etc. The power to authenticate is in many cases the power to control, and handing all authentication power to the government is beyond all reason, even if it were for well-motivated law-enforcement reasons; the security risks would be totally unacceptable.

One could imagine that Alice and Bob are merely authenticating their packets to each other, and that it is not Charles but instead a rogue law enforcement agent who is introducing the chaff, and then introducing the authenticated and chaffed message as potential justification to a judge for demanding the authentication key shared by Alice and Bob. If law enforcement had unrestricted right to plaintext, then it could demand surreptitious access to all authentication keys, even when confidentiality techniques were not being used by the participants! Again, such risks are too great to be accepted.

Similarly, a rogue law enforcement agent could introduce the chaff to Alice and Bob's authenticated packet stream, and then attempt to bring Alice and Bob to court for violating some anti-encryption or anti-confidentiality law. How can Alice and Bob defend themselves against this framing attack? They did nothing but send authenticated packets to each other! Again, this shows the difficulty (or impossibility) of drafting any kind of reasonable law restricting encryption or confidentiality technology.

It is possible to make the chaffing and winnowing technique much more efficient, allowing many bits per packet instead of just one. Here is one approach. Suppose Alice has a one-megabit message. She might pre-process the message using an ``all-or-nothing'' or ``package transform'' (Rivest 1997)---this is a keyless (non-encryption) transform that takes the message and produces a ``packaged message'' with the property that the recipient (Bob) can't produce the original message unless he has received the entire packaged message. The packaging operation can be undone by anyone who receives the packaged message; as noted, packaging is not encryption and there are no shared secret keys involved in the packaging operation. Alice might want to do so because she wants to ensure that Bob either sees all of the

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

message or none of it; he doesn't ever see just part of it. Unless the entire packaged message is received, the parts received effectively look like random noise.

Alice then breaks her packaged message into 1024-bit blocks, authenticates each block with a MAC, and transmits the result to Bob. This message is packaged and authenticated, but not encrypted: an eavesdropper can easily reconstruct the message given all of the blocks.

However, Charles can add 1024-bit chaff blocks, where each chaff block has 1024 bits of random data and a random (and presumably wrong) MAC. Again, adding the chaff provides extremely strong confidentiality, since an eavesdropper can not distinguish the chaff from the wheat. Other transforms, besides the packaging transform, might work as well.

For an adversary, the difficulty of separating the wheat blocks from the chaff will be proportional to the number of ways a subsequence of blocks can be picked as and tested for being wheat; this will be exponential in the total number of blocks, assuming that the fraction of chaff blocks is guaranteed not to be close to zero or close to one. We note that 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 deletions) in order to retrieve the message. Thus, for long messages, the relative number of chaff packets needed can be quite small, and the extra bandwidth required for transmitting chaff might be insignificant in practice.

Chaffing and winnowing bear some relationship to steganography. I am reminded of the steganographic technique of sending an innocuous-looking letter whose letters are written in two different, but very similar fonts. By erasing all letters in one font, the hidden message written in the other font, remains. For this technique (as with most steganographic techniques), security rests on the assumption that the adversary will not notice the use of two fonts. With chaffing and winnowing, the adversary may know (or suspect) that there are two different kinds of packets, but he is unable to distinguish them because he does not possess the secret authentication key.

Chaffing and winnowing also bear some resemblance to encryption techniques. Indeed, the process of authenticating packets and then adding chaff achieves confidentiality, and so qualifies as encryption by anyone who uses a definition of encryption that is so broad as to include all techniques for achieving confidentiality. But this fails to note the special structure here, wherein a non-encrypting key-dependent first step (adding authentication) followed by a non-encrypting keyless second step (adding chaff) achieves confidentiality. Since the second step can be performed by anyone (e.g. Charles in our example), and since the first step (adding authentication) may be performed for other good reasons, we see something novel, where strong confidentiality can even be obtained without the knowledge and permission of the original sender. (Variations on chaffing and winnowing, such as omitting the plaintext

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

bits altogether and letting the receiver infer them from the MAC's, destroy these nice properties.)

I note that the use of MAC's can be replaced by digital signatures. Not the ordinary kind of digital signatures, since then anyone would be able to distinguish wheat from chaff. But the recent ``designated verifier signatures'' of Jakobsson, Sako, and Impaglizzo (Jakobsson et al '96), which can only be verified by those the signer designates, would work fine. (Chaum has also independently invented the same concept.)

I note that it is possible for a stream of packets to contain more than one subsequence of ``wheat'' packets, in addition to the chaff packets. Each wheat subsequence would be recognized separately using a different authentication key. One interesting consequence of this is that if law enforcement were to demand to see an authentication key so it could identify the wheat, the sender could yield up one such key that identifies a wheat subsequence containing an innocuous message as the wheat, and leaving everything else as ``chaff''. The real message would still be buried in the chaff. This is reminiscent of the technique of ``deniable encryption'' proposed by Canetti et al. (1997).

In the chaffing and winnow approach, Alice and Bob use standard authentication techniques, and then someone adds chaff to the sequence of authenticated packets. It is worth observing that Alice and Bob can obtain a covert or subliminal channel by replacing a portion of each MAC for an ordinary message by a portion of the ciphertext for a hidden message. Without an authentication key, law enforcement cannot detect this channel. But this is outside our model.

It is also worth noting that the ability to bootstrap from authentication techniques to confidentiality mechanisms is not new. For example, two parties can use authenticated Diffie-Hellman to agree upon an encryption key. In such a case, the parties initially have only each other's signature verification keys. After the protocol is over, they have a secret shared key that they can use for encryption purposes. Chaffing and winnowing differ in that the two parties involved may not even explicitly take any steps to achieve confidentiality (if someone else is adding the chaff).

Another example of using authentication to achieve confidentiality occurs in baseball--a coach will signal to a runner by giving a sequence of signals, but the real signal is the one immediately following a previously agreed-upon authenticator signal.

A final example of using authentication to achieve confidentiality occurs in the Rex Stout's novel ``The Doorbell Rang.'' Two men wish to communicate privately, but fear that the FBI has bugged the room. They agree when the speaker raises a finger, his statements are to be disregarded. Of course, the FBI's bugs can't tell if the speaker has his finger raised or lowered!

In summary, we have introduced a new technique for confidentiality,

Chaffing and Winnowing: Confidentiality without Encryption

Ronald L. Rivest
MIT Lab for Computer Science

called ``chaffing and winnowing''. This technique can provide excellent confidentiality of message contents without involving encryption or steganography. As a consequence of the existence of chaffing and winnowing, one can argue that attempts by law enforcement to regulate confidentiality by regulating encryption must fail, as confidentiality can be obtained effectively without encryption and even sometimes without the desire for confidentiality by the two communicants. Law enforcement would have to seek access to all authentication keys as well, a truly frightening prospect.

Mandating government access to all communications is not a viable alternative. The cryptography debate should proceed by mutual education and voluntary actions only.

Acknowledgments

Thanks to my dad for suggesting the term ``winnowing,' ' to Mark Lomas for noting that multiplexing two streams may allow each to serve as chaff for the other, and to Peter Wayner for suggesting the relationship to deniable encryption. Thanks to Adi Shamir and David Gifford for suggesting the basic idea underlying the more efficient implementation of chaffing and winnowing; Aaron Gifford first noted that the number of chaff packets might be small in this case. Thanks also to Matt Blaze and Markus Jakobsson for comments on the original write-up. And finally thanks to Bruce Balden and Enzo Michelangeli for bringing the Rex Stout reference to my attention.

References

Canetti, Ran, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky, "Deniable Encryption", Proceedings CRYPTO '97 (Springer 1997), 90--104.
<ftp://theory.lcs.mit.edu/pub/tcryptol/96-02r.ps>

Jakobsson, Markus, Kazue Sako, and Russell Impagliazzo, ``Designated Verifier Proofs and Their Applications'', Proceedings Eurocrypt '96 (Springer 1996), 143--154.
<http://www.bell-labs.com/user/markusj/dvp.ps>

Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC2104, February 1997.
(Available at <ftp://ds.internic.net/rfc/rfc2104.txt>)

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

Stout, Rex. The Doorbeel Rang: A Nero Wolfe Novel. (Viking Press, 1965).

Wayner, Peter. Disappearing Cryptography: Being and Nothingness on the Net. Academic Press, 1996.