***Binary-to-text encoding***

It is often desirable, to be able to send non-textual data through text-based systems, such as when one might attach an image file to an e-mail message. To accomplish this, the data is encoded in some way, such that eight-bit data is encoded into seven-bit ASCII characters (generally using only alphanumeric and punctuation characters-the ASCII printable characters). Upon safe arrival at its destination, it is then decoded back to its eight-bit form. This process is referred to as **binary-to-text encoding**. Many programs perform this conversion to allow for data-transport, such as PGP and GNU Privacy Guard (GPG).

A binary-to-text encoding is encoding of data in plain text. More precisely, it is an encoding of binary data in a sequence of printable characters. These encodings are necessary for transmission of data when the channel does not allow binary data (such as email or NNTP) or is not 8-bit clean. PGP documentation (RFC 4880) uses the term ASCII armor for binary-to-text encoding when referring to Base64.

**Encoding plain text**

However, **binary-to-text encoding** methods are also used as a mechanism for encoding plain text. For example:

* Some systems have a more limited character set they can handle; not only are they not 8-bit clean, some cannot even handle every printable ASCII character.
* Other systems have limits on the number of characters that may appear between line breaks, such as the "1000 characters per line" limit of some SMTP software, as allowed by RFC 2821.
* Still others add headers or trailers to the text.
* A few poorly-regarded but still-used protocols use in-band signaling, causing confusion if specific patterns appear in the message. The best-known is the string "From " (including trailing space) at the beginning of a line used to separate mail messages in the mbox file format.

By using a binary-to-text encoding on messages that are already plain text, then decoding on the other end, one can make such systems appear to be completely transparent. This is sometimes referred to as 'ASCII armoring'. For example, the ViewState component of ASP.NET uses **base64** encoding to safely transmit text via HTTP POST, in order to avoid delimiter collision.