DANIELMIESSLER

Encoding vs. Encryption vs. Hashing vs. Obfuscation

By DANIEL MIESSLER (HTTPS://DANIELMIESSLER.COM/BLOG/AUTHOR/DANIEL/)

CREATED/UPDATED: MAY 7, 2019

```
<script
type="text/javascript">document.write('\u003C\u0064\u0069\u0076\u
0020\u0073\u0074\u0079\u006C\u0065\u003D\u201D\u0076\u0069\u0073\
u0069\u0062\u0069\u006C\u0069\u0074\u0079\u003A\u0020\u0068\u0069
\u0064\u0064\u0065\u006E\u003B\u0020\u0070\u006F\u0073\u0069\u007
4\u0069\u006F\u006E\u003A\u0020\u0061\u0062\u0073\u006F\u006C\u00
75\u0074\u0065\u003A\u0020\u0031\u003B\u0020\u0074\u006F\u0070\u0
03A\u0020\u0031\u201D\u003E\u0020\u000D\u003C\u0069\u0066\u0072\u
0061\u006D\u0065\u0020\u0069\u0064\u003D\u0022\u0049\u0046\u0052\
u0041\u004D\u0045\u0022\u0020\u006E\u0061\u006D\u0065\u003D\u0022
\u0049\u0046\u0052\u0041\u004D\u0045\u0022\u0020\u0073\u0072\u0066
3\u003D\u0022\u0068\u0074\u0074\u0070\u003A\u002F\u002F\u0077\u00
77\u0077\u002E\u0065\u0078\u0061\u006D\u0070\u006C\u0065\u002E\u0
063\u006F\u006D\u002F\u0070\u0061\u0067\u0065\u005F\u0077\u0069\u
0074\u0068\u005F\u006D\u0061\u006C\u0077\u0072\u0065\u002E\u0068\
u0074\u006D\u0022\u0020\u0073\u0063\u0072\u006F\u006C\u006C\u0069
\u006E\u0067\u003D\u0022\u006E\u006F\u0022\u0020\u0077\u0069\u0066
4\u0074\u0068\u003D\u0022\u0031\u0022\u0020\u0068\u0065\u0069\u00
67\u0068\u0074\u003D\u0022\u0031\u0022\u0020\u0076\u0073\u0070\u0
061\u0063\u0065\u003D\u0030\u0020\u0068\u0073\u0070\u0061\u0063\u
0065\u003D\u0030\u0020\u0066\u0072\u0061\u006D\u0065\u0062\u006F\
u0072\u0064\u0065\u0072\u003D\u0022\u0030\u0022\u003E\u003C\u002F
\u0069\u0066\u0072\u0061\u006D\u0065\u003E\u003C\u002F\u0064\u006
9\u0076\u003E\u000D');</script>
```


There is often SIGNIFICANT CONFUSION (HTTPS://WWW.GOOGLE.COM/SEARCH?SOURCEID=CHROME&IE=UTF-8&Q=WHAT%27S+THE+DIFFERENCE+BETWEEN+ENCODING+AND+ENCRYPTION%3F& around the differences between encryption, encoding, hashing, and obfuscation.

> GET THE TL;DR

Let's take a look at each one:

Encoding

000d	*00h*		(nul)	*016d*	*10h*	►	(dle)
001d	*01h*	☺	(soh)	*017d*	*11h*	◄	(dc1)
002d	*02h*	☻	(stx)	*018d*	*12h*	↕	(dc2)
003d	*03h*	♥	(etx)	*019d*	*13h*	‼	(dc3)
004d	*04h*	♦	(eot)	*020d*	*14h*	¶	(dc4)
005d	*05h*	♣	(enq)	*021d*	*15h*	§	(nak)
006d	*06h*	♠	(ack)	*022d*	*16h*	▬	(syn)
007d	*07h*	•	(bel)	*023d*	*17h*	↨	(etb)
008d	*08h*	◘	(bs)	*024d*	*18h*	↑	(can)
009d	*09h*		(tab)	*025d*	*19h*	↓	(em)
010d	*0Ah*		(lf)	*026d*	*1Ah*		(eof)
011d	*0Bh*	♂	(vt)	*027d*	*1Bh*	←	(esc)
012d	*0Ch*	♀	(np)	*028d*	*1Ch*	∟	(fs)
013d	*0Dh*		(cr)	*029d*	*1Dh*	↔	(gs)
014d	*0Eh*	♪	(so)	*030d*	*1Eh*	▲	(rs)
015d	*0Fh*	☼	(si)	*031d*	*1Fh*	▼	(us)

The purpose of *encoding* is to transform data so that it can be properly (and safely) consumed by a different type of system, e.g. binary data being sent over email, or viewing special characters on a web page. The goal is **not** to keep information secret, but rather to ensure that it's able to be properly consumed.

Encoding transforms data into another format using a scheme *that is publicly available* so that it can easily be reversed. It does not require a key as the only thing required to decode it is the algorithm that was used to encode it.

Examples: ASCII (HTTP://WWW.ASCIITABLE.COM/), UNICODE (HTTPS://DANIELMIESSLER.COM/STUDY/ENCODING/#UNICODE), URL Encoding, BASE64 (HTTPS://EN.WIKIPEDIA.ORG/WIKI/BASE64)

Encryption

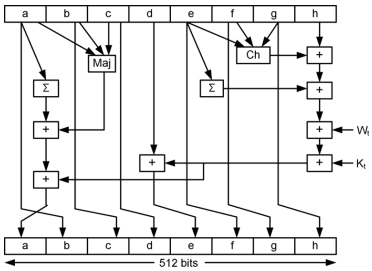

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.5 (GNU/Linux)

hQIOAOuHn1ue4n32EAf/UEF6JLrap1OBMdKMvb+Dz9GvoijUixH+gbcpi9qGa+43
vC3ktMwo7OWqPyJseVRSPBOv6dOwy65KrzrHwhOHO/CKEk2O5STAwzj6C3USgDfZ
6E+Gc4iumMl725JNahJzcL5ED33LFdZ6uoEjgqggxG1dFwvwksRHA4+VU9Bcd5eL
T9aRVbkXNxXkQn2FWhWuhPQFNWLwIVrDd9TPtDvpRTl6YiB1AM9ks3HlYZHL7mfR
Hk9yfy1nGXdhiO6EDvvTvd/Lq1xsFjKh6y/pG6NxABGdT6VoeWGVtQGqwpbOZGgq
xoSYkWm8MmAkkqYXZLraSEzyxxxu4cQzvzz3vrpN3AgAhObP2eUFU29EJAQpdKJW
fKAhohPVpd6+ETnzL53VLg1IJJdNGlpIziO9alNnYmDSnt2EwAELqTUl3jPiGYt5
cvSUBe3ER4/CkjvYXOVaO7ezHmCAkQpB2ILV8OwI74DQn7tNKf2gJnwzkYAF7yyf
XFGlJ8oaLpRV499mN7lNfo+ZV2HrR9xti+jUPFv+H+ROt4fMmAU5I95UksQFe/A9
YUdSBAEqKkW9zLDgpWS2oxJymGufBdhzxpw7uJlzrwsHIYIt7PSeJG4VO+xJqHvO
1qHXSukK648FlOImmVUM9csPOcvfOMZeAgh4i+HYQvFF/kGHp6ogevD4pVhztbzd
F9JhAbJSeOvZKZFPhzjgX+mCgvzVRniSdDg7wc3+YKNei2zQrmTsiiO6JyhQV2OI
tAqTk572zdZbrCtSgcthrN/uxbJSNnw4X9IZbWtFOUr3lr676II8Q112ttO3IVCe
fF/pZA==
=sPWf
-----END PGP MESSAGE-----
```

The purpose of *encryption* is to transform data in order to keep it secret from others, e.g. sending someone a secret letter that only they should be able to read, or securely sending a password over the Internet. Rather than focusing on usability, the goal is to ensure the data cannot be consumed by anyone other than the intended recipient(s).

Encryption transforms data into another format in such a way that *only specific individual(s)* can reverse the transformation. It uses a key, which is kept secret, in conjunction with the plaintext and the algorithm, in order to perform the encryption operation. As such, the ciphertext, algorithm, and key are all required to return to the plaintext.

Examples: AES (http://www.aes.org/), Blowfish (https://en.wikipedia.org/wiki/Blowfish_(cipher)), RSA (https://www.rsa.com/)

Hashing



Hashing serves the purpose of ensuring *integrity*, i.e. making it so that if something is changed you can know that it's changed. Technically, hashing takes arbitrary input and produce a fixed-length string that has the following attributes:

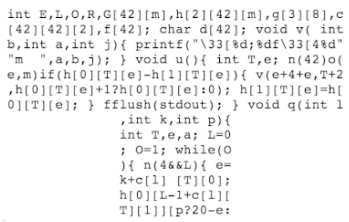
1. The same input will always produce the same output.
2. Multiple disparate inputs should not produce the same output.
3. It should not be possible to go from the output to the input.
4. Any modification of a given input should result in drastic change to the hash.

Hashing is used in conjunction with authentication to produce strong evidence that a given message has not been modified. This is accomplished by taking a given input, hashing it, and then signing the hash with the sender's private key.

When the recipient opens the message, they can then validate the signature of the hash with the sender's public key and then hash the message themselves and compare it to the hash that was signed by the sender. If they match it is an unmodified message, sent by the correct person.

Examples: SHA-3 (HTTPS://EN.WIKIPEDIA.ORG/WIKI/SHA-3), MD5 (NOW OBSOLETE) (HTTPS://EN.WIKIPEDIA.ORG/WIKI/MD5), ETC. (HTTPS://EN.WIKIPEDIA.ORG/WIKI/MD5)

Obfuscation



The purpose of obfuscation is to make something harder to understand, usually for the purposes of making it more difficult to attack or to copy.

One common use is the the obfuscation of source code so that it's harder to replicate a given product if it is reverse engineered.

It's important to note that obfuscation is not a strong control (like properly employed encryption) but rather an obstacle. It, like encoding, can often be reversed by using the same technique that obfuscated it. Other times it is simply a manual process that takes time to work through.

Another key thing to realize about obfuscation is that there is a limitation to how obscure the code can become, depending on the content being obscured. If you are obscuring computer code, for example, the limitation is that the result must still be consumable by the computer or else the application will cease to function.

Examples: JAVASCRIPT OBFUSCATOR (HTTPS://JAVASCRIPTOBFUSCATOR.COM), PROGUARD (HTTPS://WWW.GUARDSQUARE.COM/EN/PRODUCTS/PROGUARD)

Summary

- **Encoding** is for maintaining data *usability* and can be reversed by employing the same algorithm that encoded the content, i.e. no key is used.
- **Encryption** is for maintaining data *confidentiality* and requires the use of a key (kept secret) in order to return to plaintext.
- **Hashing** is for validating the *integrity* of content by detecting all modification thereof via obvious changes to the hash output.
- **Obfuscation** is used to *prevent people from understanding* the meaning of something, and is often used with computer code to help prevent successful reverse engineering and/or theft of a product's functionality.

Notes

1. One might ask when obfuscation would be used instead of encryption, and the answer is that obfuscation is used to make it harder for one entity to understand (like a human) while still being easy to consume for something else (like a computer). With encryption, neither a human or a computer could read the content without a key.