cloudwards.net

What Is AES Encryption & How Does It Work in 2022? [256-bit vs 128-bit]

Aleksander Hougen

14-17 minutes

Cloudwards.net may earn a small commission from some purchases made through our site. However, any affiliate earnings do not affect how we review services.

Table of Contents

- What Is AES Encryption? Breaking Down the Basics
- How AES Encryption Works
- Key Size: 256-bit vs 192-bit vs 128-bit
- Battling Encryption: AES vs RSA
- AES Security Issues
- Final Thoughts

- Comments

  If you've done research into any kind of consumer software — whether it's cloud storage, backup providers or virtual private networks — you've likely come across the phrase "AES." You generally don't get an explanation beyond that, though. So, what is AES encryption exactly, and how does it work? Join us as we break down the popular encryption protocol to its core components in an effort to decipher some of the technobabble surrounding encryption.

  Key Takeaways:

- AES stands for "Advanced Encryption Standard."
- The AES algorithm is the industry-standard encryption protocol that protects sensitive information from traditional brute-force attacks.
- The two most common versions are 256-bit AES (providing greater security) and 128-bit AES (providing better performance during the encryption and decryption process).
- With current technology, AES is uncrackable through straightforward, brute-force attacks, and it is used in countless applications, from protecting top-secret or classified information in government agencies to keeping your personal data safe when stored on the cloud.

  Since its adoption in 2000 as the industry standard, AES has become ubiquitous in every part of our digital lives. Unless you've been living in a cave for the last 20 years, you're pretty much guaranteed to have used software that uses it, even if you have no idea what it is.

- Short for "Advanced Encryption Standard," AES is the most widely used protocol to encrypt data and keep it safe from prying eyes. How it works exactly is too complicated for a short answer, so keep reading the article if you'd like to learn more.
- AES encryption is used in practically everything computer-related:

from consumer-facing products, such as cloud storage or VPNs, to massive government and corporate systems that require a high degree of certainty that files remain private and secure.

- Encryption allows us to use computers and the internet to perform tasks and store sensitive data that needs to be confidential. For example, without encryption, things like your banking details, medical records and even purchases would be far too exposed to exist safely in a digital space.

What Is AES Encryption? Breaking Down the Basics

As is the case with walking before running, to understand what AES is, we need to explain some basics about encryption and how AES encryption came to be.

What Is Encryption?

Encryption is a wide topic that takes years to fully understand, so we won’t take up too much space in this guide to explain the fundamental principles of the practice.

In short, though, encryption generally works by taking a piece of data you want to keep confidential and scrambling it, making it unreadable to anyone who doesn’t have the secret key required to unscramble it.

Different protocols go about this differently, with some — such as AES — utilizing a “symmetric-key algorithm,” which means that the same key encrypts and decrypts the data.

The opposite of symmetric encryption is (surprise, surprise) “asymmetric encryption,” where a publicly available key encrypts the data. However, a separate secret private key has to decrypt it again.

If you need a more in-depth explanation of the basics of encryption, head over to our general description of encryption before continuing.

The History of AES

As we've already mentioned, AES simply stands for "Advanced Encryption Standard," a name it adopted when it was chosen as the industry standard for encryption by the U.S. National Institute of Standards and Technology (NIST, for short). Originally named Rijndael, the encryption was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, in the mid-to-late 1990s.

Along with 14 other protocols — including the four that would go on to compete with Rijndael: namely, Twofish, Serpent, RC6 and MARS — it was submitted to NIST for consideration as the new industry standard for encryption. After three summits in 1998, 1999 and 2000, NIST announced that Rijndael was the winner and would henceforth be known as AES.

Now that we've given a brief top-level explanation of what encryption does, as well as a quick history of the protocol, it's time to dig into the specifics of AES encryption.

We'll do our best to explain this as simply as possible, but you might still feel a bit lost among the references to substituting bytes, shifting rows and mixing columns. If you do, don't worry, encryption is a notoriously complex thing to understand, and you shouldn't expect to be a master on the subject from reading a simplified explanation of the process.

At its most basic level, the AES encryption process consists of four different stages. We'll explain each of these as we go through the encryption algorithm step by step, but in order, they are: add round key, sub bytes and mix columns.

As we go through the steps of the encryption process, we need some examples. Let's say that the data we want to encrypt is a simple phrase in plaintext, "encryption test1," and the encryption key is "password12345678."

This encryption key is just an easy example, and you should **never**

use this as an actual key; instead, create strong passwords with long and complicated key lengths. You can even use our password generator to create very strong encryption keys.

Step 1: Dividing the Data

Before AES can do anything, it needs to take the to-be-encrypted data and transform it into blocks that it will then encrypt in batches. This is done by taking the input data — for example, plaintext — and cutting it into pieces.

These blocks are structured into **arrays** (think tables) composed of columns and rows in a four-by-four pattern. Since each "cell" in the array consists of a single byte, you end up with a block size of 128 bits (there are eight bits in a byte, so eight multiplied by 16 gives us 128 bits).

So, using our example data of "encryption test1," we get an array that looks like this:

e	y	o	e
n	p	n	s
c	t		t
r	i	t	1

Step 2: Key Expansion

Next, AES takes your key (for example, a password) and places it into the same block structure used for the data you're encrypting. It then replaces each part of the key with what appears to the naked eye as a jumble of random numbers and letters.

Before the replacement, our key looks like this:

p	w	1	5
a	o	2	6
s	r	3	7

s	d	4	8

Once this runs through the key expansion process, it turns into multiple blocks, starting with this one:

70	77	31	35
61	6f	32	36
73	72	33	37
73	64	34	38

The expanded key is far longer than this, but we'll just use the first block here for the sake of simplicity. This expanded key isn't used again until the final step of the round.

While these symbols may seem random, they're pulled from a substitution table known as the "AES key schedule," which has a predetermined replacement value for every symbol. If you are curious about the "6f" in the blocks, this is a hexadecimal, not a typo, and you'll see more of these in the process.

Step 3: Add Round Key

Finally, we get to the actual AES encryption process. The protocol now uses the blocks of data created in "step one" and combines them with your original key to create an entirely new block cipher using the key schedule. While adding text like they're numbers might seem strange, it's worth remembering that to a computer, everything is a number, including text.

So, once this process is complete, it leaves us with a block that looks like this:

15	0e	5e	50
0f	1f	5c	45
10	06	13	43
01	0d	40	09

Step 4: Byte Substitution

Using the block cipher created in “step three,” AES now replaces every single byte in the cypher with something else according to a table known as the Rijndael S-box. Once this is complete, our block of symbols has changed to this:

59	ab	58	53
76	c0	4a	6e
ca	6f	7d	1a
7c	d7	09	01

We’ve marked each column with a specific color here so that you can more easily follow what happens in the next step.

Step 5: Shifting Rows

To further scramble the data created in our previous four steps, AES then shifts the rows to the left. The first row stays the same, but rows two, three and four are then shifted to the left by one, two and three bytes, respectively. That leaves us with this:

59	ab	58	53
c0	4a	6e	76
7d	1a	ca	6f
01	7c	d7	09

As you can tell by the colors, the rows have now shifted.

Step 6: Mixing Columns

The next step of the process takes each column of the block ciphers and runs them through a series of complicated mathematics that produces an entirely new set of columns. The precise math is far too complicated to explain here, but what you’re left with are blocks of data that now look completely different:

95	f5	1f	5a
44	6d	16	07
60	51	db	e0
54	4e	f9	fe

Step 7: Add Round Key

In the final step of the encryption process, AES takes the key created in “step two” (key expansion) and adds it to the block cipher we ended up with at the end of “step six.” This changes our cipher yet again to the following:

e1	f6	2d	5d
bf	f9	b0	97
14	57	ee	e2
b1	cf	4c	73

Step 8: Repeating Previous Steps

Once all seven steps are complete, steps four to seven are repeated a number of times, with the exact number of rounds determined by the key size used. If the key size is 128 bits, then AES goes through 10 rounds, as opposed to 12 rounds for the 192-bit and 14 rounds for the 256-bit. The final round skips step six, but otherwise each repetition is the same.

Each round further complicates the encryption, making it harder and harder to distinguish from its original state. That is, unless you have access to the original key, which can essentially perform the same process in reverse to decrypt the data.

Key Size: 256-bit vs 192-bit vs 128-bit

As we mentioned earlier, AES is an overarching term that covers the entire Rijndael family of encryption protocols. This group consists of three different algorithms that are mostly the same, with one significant exception: the key size used.

There are three different sizes: 256-bit AES, 192-bit AES and 128-bit AES. The largest size, 256-bit AES, is the most secure, while 128-bit is conversely the least secure of the three. That said, all three key sizes are strong enough to repel even the most dedicated brute-force attack, but the two smaller key sizes are theoretically easier to crack (the time it would take to crack 128-bit AES through a brute-force attack is still billions of years).

The middle-grade 192-bit keys are seldom used, with most AES encryption and decryption using either a 256-bit key or 128-bit key. Although 256-bit encryption is more secure, a 128-bit key uses less computing power. Thus, it's useful for less sensitive data or when the encryption process has limited resources available to it.

Battling Encryption: AES vs RSA

If you've heard the term "AES"' before, you might also have heard of another famous encryption protocol: RSA. An acronym for its three inventors (Rivest, Shamir and Adelman), RSA is even more secure than AES, mostly due to the fact that it uses an asymmetric key model rather than a symmetric one.

So, if RSA exists and it's even more secure than AES, it begs the question of why we even use AES in the first place. The answer is simple enough, as it's due to the way it handles encryption: RSA requires significantly more computing power than AES, and as such, is not suitable for applications where performance and speed are critical.

Although a symmetric block cipher will never be as secure as an asymmetric one, its relatively low-performance requirements mean that it's an ideal encryption algorithm for hardware and software that need to encrypt and decrypt data quickly and efficiently without meaningfully compromising on security.

AES Security Issues

There are theoretical weaknesses to AES, although it is a remarkably safe encryption protocol (it would take billions of years even for organizations with tons of computing power, such as the National Security Agency).

However, while simple brute-force attacks have never been able to crack AES encryption, if AES is poorly implemented, it can be targeted in other more complicated types of attacks.

The first way to exploit a poorly implemented AES algorithm is through something called a “related-key attack.” This type of attack is possible if the attacker has some way of linking the known public key to the corresponding secret private key; thus, gaining the ability to decrypt the data. If the AES-key generation is implemented properly, though, this method of attack is not feasible.

Side-Channel Attacks

Computers are essentially just a bunch of electrical signals, which, in theory, can be picked up by monitoring things like physical electromagnetic leaks. If these leaks contain the private key, the attacker can then decrypt the data as if they were the intended receiver.

Known-Key Distinguishing Attacks

A known-key attack is probably the simplest to understand, as it requires the attacker to know the key used to encrypt the data. With the key in hand, the attacker can compare the encrypted data to other encrypted data where the contents are known to decipher it.

However, this type of attack is only effective against seven rounds of AES encryption, which means that even the shortest key length (128-bit) would be immune, as it uses 10 rounds. Additionally, the likelihood of an attacker knowing the original key is very low.

Key-Recovery Attacks

A key-recovery attack means that an attacker has their hands on a piece of encrypted and decrypted data and can then use that information to deduce what the key used to encrypt it was. Luckily, this is a theoretical avenue of attack, as it’s only four times faster than a brute-force attack, which means that they’d still have to spend billions of years on it.

Final Thoughts: AES Encryption

Now that you’ve reached the end of our guide, we hope we’ve provided you with a satisfying answer to the question of what AES encryption is. If you’re still a bit confused, that’s understandable, as cryptography is an inherently complex field that requires years of study to truly understand on a fundamental level.

What did you think of our crash course on AES encryption? Do you feel you have a better understanding of how the protocol protects your data, or are you just as lost as you were before reading? Let us know in the comments below, and as always, thank you for reading.