AES-256 is the standardized encryption specification. It's used worldwide by everyone from local wireless encryption at home, to the US government, and to major corporations. The key sizes for AES are 128, 192, and 256 bits. This means that in its strongest from, using a 256 bit key, there are 2^{256} possible keys to try if you are considering a worst case brute force solution. In key cracking, we normally expect to have to try only half the keyspace, which is then 2^{255}. We will call this number J. The "work factor" question, for AES 256 is as always, "How long would it take to test each of the possible keys?" In the following discussion, we will use the term "flops" which is an abbreviation for "floating point operations per second." Key cracking takes floating point operations.

Graphical Processing Units (GPUs) are better suited for this type of computing than CPUs. A high-end GPU can typically do about 2 billion calculations per second. If you hooked a billion such GPUs together into a massively parallel machine, they would then be theoretically able to do 1 billion * 2 billion flops. This works out to **$2 * 10^{18}$ flops** for this massively parallel GPU based machine. **We will call this huge number K.**

To calculate the number of seconds in a year, we multiply 60 seconds/minute * 60 minutes/hour * 24 hours/day * 365 days/year. This yields 31,556,952 seconds in a year. **We will call this number L.** The product, K * L, gives the number of keys per year that this massively parallel machine can test. This number is a little more than $6.3 * 10^{25}$. We will call this number M. To determine how many years it will take to test half the AES 256 keyspace, we divide the total number of keys, J, by the number of keys per year, M. To simplify matters, we will use L as 6.3, so when we calculate 2^{255} / L which is more than $9 * 10^{50}$ years.

In tablular form, this is what we have roughly determined

Number of AES 256 Keys	**Number of keys for second for our massively parallel machine**	**Number of keys for year for our massively parallel machine**
Letter J or 2^{255}	**Letter K or $2 * 10^{18}$**	**Letter L or $6.3 * 10^{25}$**
Letter M is the	**M = J / L =**	

Number of Years to find AES Key	**$9 * 10^{50}$**	

To add some physics to the mix, **the best estimates that scientists have is that our universe has only existed for 14 billion, or $1.4 * 10^9$ years.** We will call this number N. Then, dividing M by N, we find that it would take over $6 * 10^{40}$ times longer than the age of our universe itself to exhaust half of the keyspace of an AES-256 key.

On top of this, there is an energy limitation. Most high-end GPUs take around 150 watts of energy to power themselves at full load, not including the power to run the required cooling systems. Multiplying all this together, we conclude that to power this machine takes 150 watts per GPU * 1 billion GPUs = $1.50 *10^{11}$ watts to simultaneously run our "massively parallel machine." We call this number O "short for OMG or whatever you want to call it." Power estimates suggest this amount of power will simultaneously power 50 million American households. **This one is a wowser!!**

The largest nuclear power reactors are found in Japan, and they generate about 1 gigawatt or 10^9 watts of energy. Since there are O watts required, this means we would require 150 nuclear power plant reactors to constantly power our massively parallel AES key buster machine, and it would still take significantly longer than the age of the universe to exhaust half of an AES-256 keyspace.

But wait? **There are supercomputers out there.** Maybe they could solve this problem for us. The Chinese **Tianhe-2** is currently the world's fastest supercomputer. It cranks out around 34 petaflops or $34 * 10^{15}$ flops. Doing the math for this basketball court sized supercomputer, we find it can test about $5.4 * 10^{52}$ AES 256-bit keys per second. Without going through all the same mathematical steps as before, we come to a startling conclusion. **It would take 10^{38} Tianhe-2 Supercomputers running for the entirety of the existence of everything to exhaust half of the keyspace of an AES-256 key.**

Of course, there is always the **Moore's Law argument** about the number of transistors on a CPU chip doubling each 18 months. But we are so far off from solving this problem, that maybe NSA and NIST were both right when they introduced the AES competition in 1997. The subtitle for AES was "The Advanced Encryption Algorithm: **Encryption for the next Century!"**