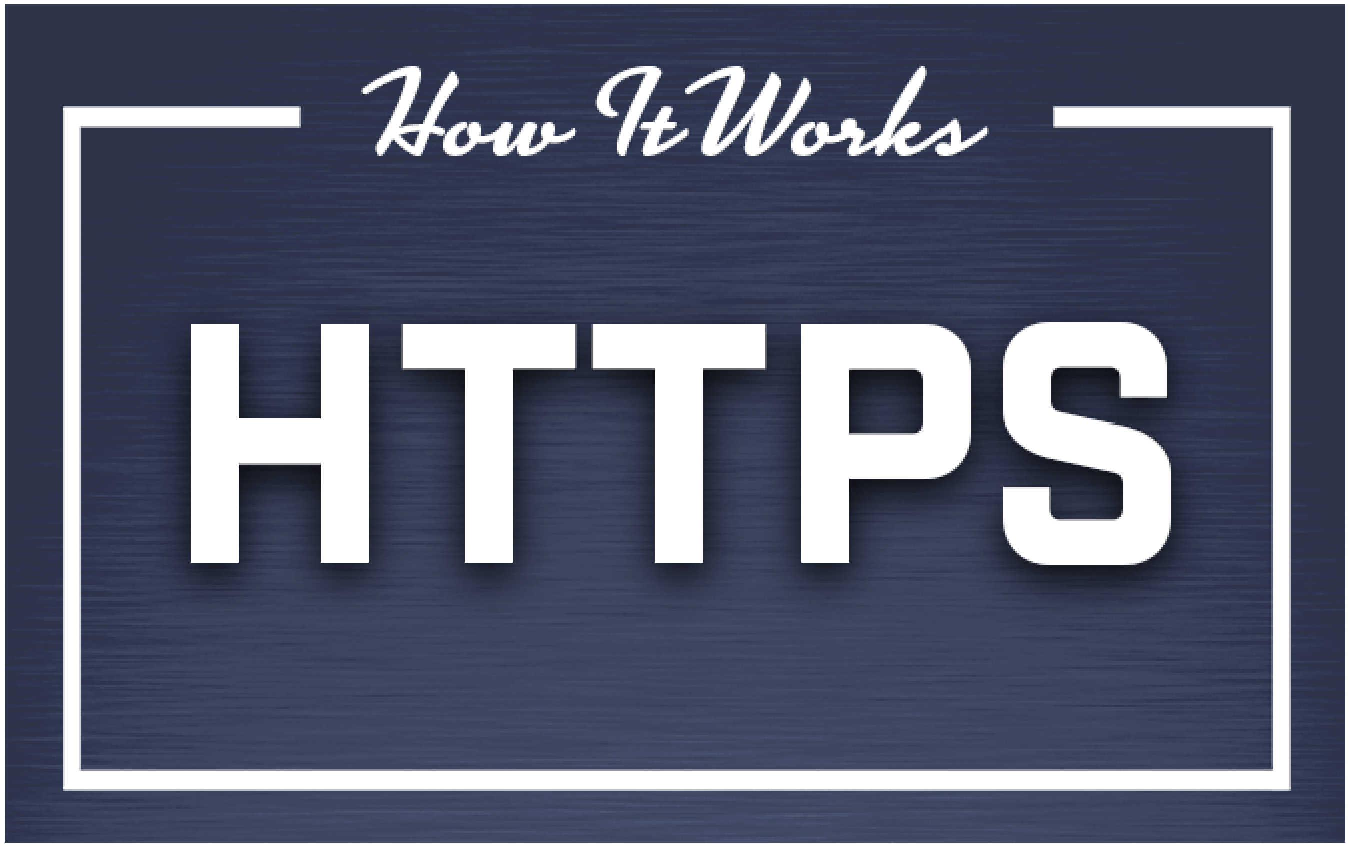
How Does HTTPS Work? RSA Encryption Explained



10 Sep, 2017 No Comments Bobby How Security Works

TipTopSecurity has finally been transitioned to a fully HTTPS website!

Secure | https://tiptopsecurity.com

So naturally, I thought this would be the perfect time to explain what that means. Read on for a complete explanation.

Note: This article explains the older RSA encryption method. The newer ECC method is arguably better, however RSA is still more widely deployed for several reasons. Stay tuned for when I have my ECC explanation posted.

Expand Table of Contents

What is HTTPS?

In 1990, the internet as we know it was born. Since the beginning, it has used the HyperText Transfer Protocol (HTTP) for moving information around the world. That's why the beginning of web addresses start with HTTP.

Plain old HTTP is not secure because it transports information in *plain text*. This means that anyone who intercepts the traffic can read it. That includes not only the hacker who's monitoring the coffee shop's WiFi, but your internet service provider (ISP) as well. Kind of like a switchboard operator can listen in on phone calls.

But people soon decided they wanted to use the internet for sensitive data (like credit card numbers), so we had to figure out a way to make HTTP secure so that no one could see your credit card number as it zoomed between your browser and the web server.

POPULAR ARTICLES

- The Safest Web Browser
- Wipe Your Hard Drive
- What is Encryption
- How to Use AxCrypt
- How to Use SpiderOak

ADVERTISEMENT

So in 1994, Netscape Communications enhanced HTTP with some encryption. Essentially, they married a new encryption protocol named Secure Socket Layer (SSL) to the original HTTP. This became known as “HTTP over SSL” or “HTTP Secure”. Otherwise known as HTTPS.

Today, more than 50% of all websites are HTTPS. That number has been growing radically in the last few years since Edward Snowden revealed that the NSA is spying on everyone’s internet traffic.

The idea, as stated by many, is to migrate the entire internet into a completely HTTPS environment, where all website traffic is encrypted by default.

Why encrypt the entire internet?

HTTPS does as much for privacy as for security. It’s one thing to keep hackers from reading your data or injecting their own code into your web sessions (which HTTPS prevents), but privacy is the other side of the coin.

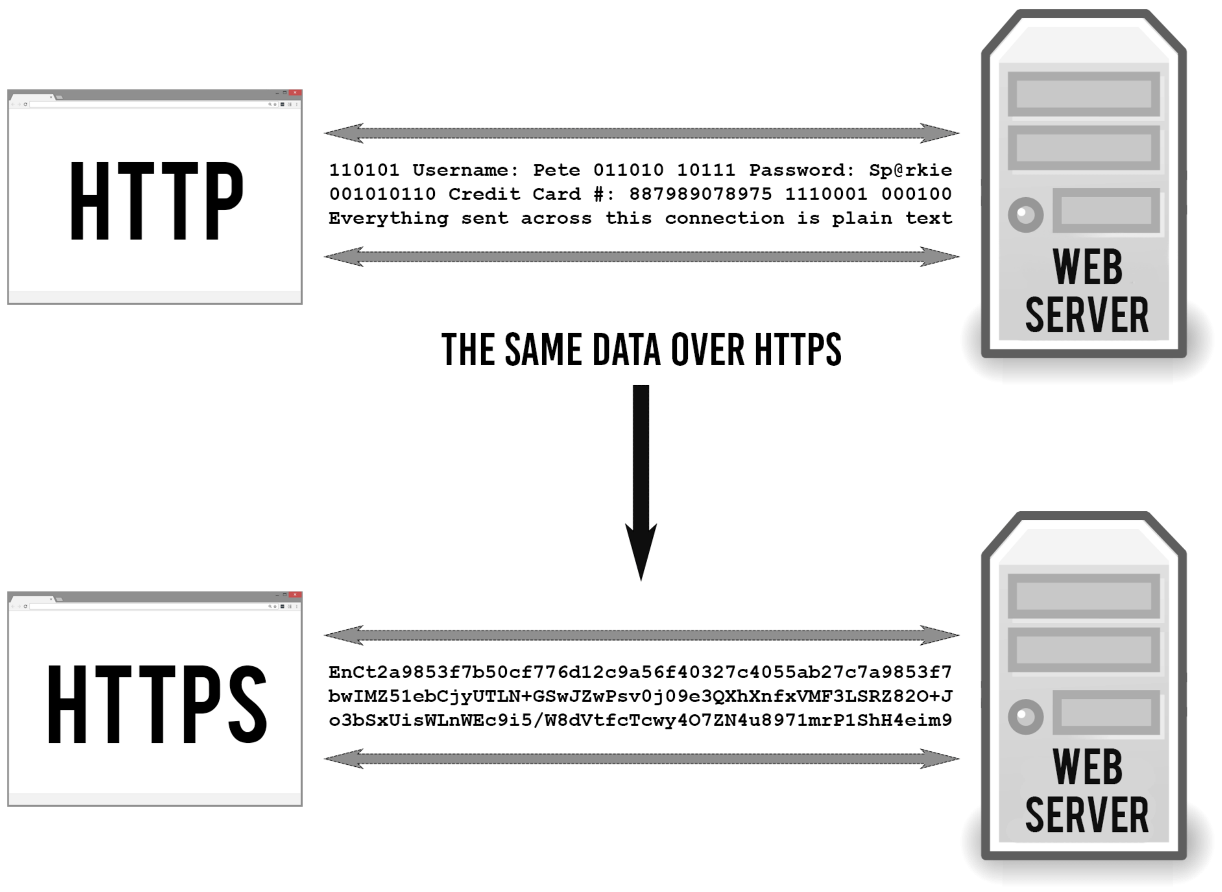
We know that ISPs, governments and big data collection firms just love snooping on and storing our traffic for God-knows-what. Sure, you may not think you care. That is, until you’re surfing information on a personal medical condition or advice on teen pregnancy. Whose business is that? That information is always useful to someone, which is why they want it and keep it. Forever.

This is why many websites (like TipTopSecurity) choose to encrypt your traffic even though you’re not sending sensitive information. Because we believe that your behavior online should remain as private as possible.

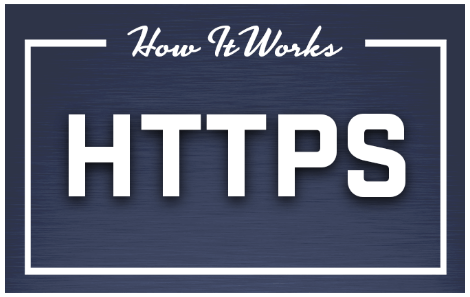
How HTTPS Works

LATEST ARTICLES

HTTPS keeps your stuff secret by encrypting it as it moves between your browser and the website's server. This ensures that anyone listening in on the conversation can't read anything. This could include your ISP, a hacker, snooping governments, or anyone else who manages to position themselves between you and the web server.

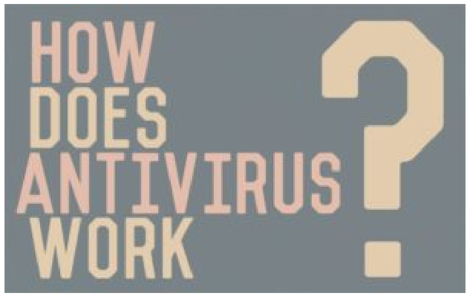

HTTP
110101 Username: Pete 011010 10111 Password: Sp@rkie
001010110 Credit Card #: 887989078975 1110001 000100
Everything sent across this connection is plain text
WEB SERVER
THE SAME DATA OVER HTTPS
HTTPS
EnCt2a9853f7b50cf776d12c9a56f40327c4055ab27c7a9853f7
bwIMZ51ebCjyUTLN+GSwJZwPsv0j09e3QXhXnfxVMF3LSRZ82O+J
o3bSxUisWLnWEc9i5/W8dVtfcTcwy4O7ZN4u8971mrP1ShH4eim9
WEB SERVER

For a long time, SSL was the standard protocol used by HTTPS. The newest version of SSL is now called Transport Layer Security (TLS) but they are essentially the same thing. I'll refer to it from now on as SSL/TLS since both monikers are used interchangeably, but technically I'm talking about the newer TLS.



Encryption Explained

What is the Most Secure Web Browser? Comparison of the 6 Most Popular Browsers



How Does Antivirus Work?

All About VPNs

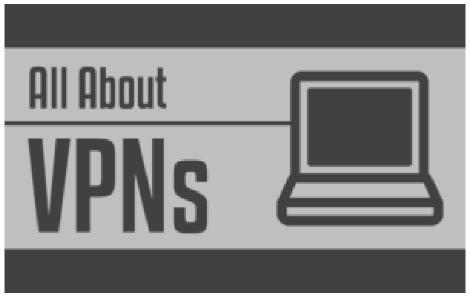
Essentially, you need three things to encrypt data:

1. The data you want to encrypt
2. A unique encryption key (just a long string of random text)
3. An encryption algorithm (a math function that "garbles" the data)

You plug the data and the key into the algorithm and what comes out the other side is *cipher text*. That is, the encrypted form of your data which looks like gibberish.

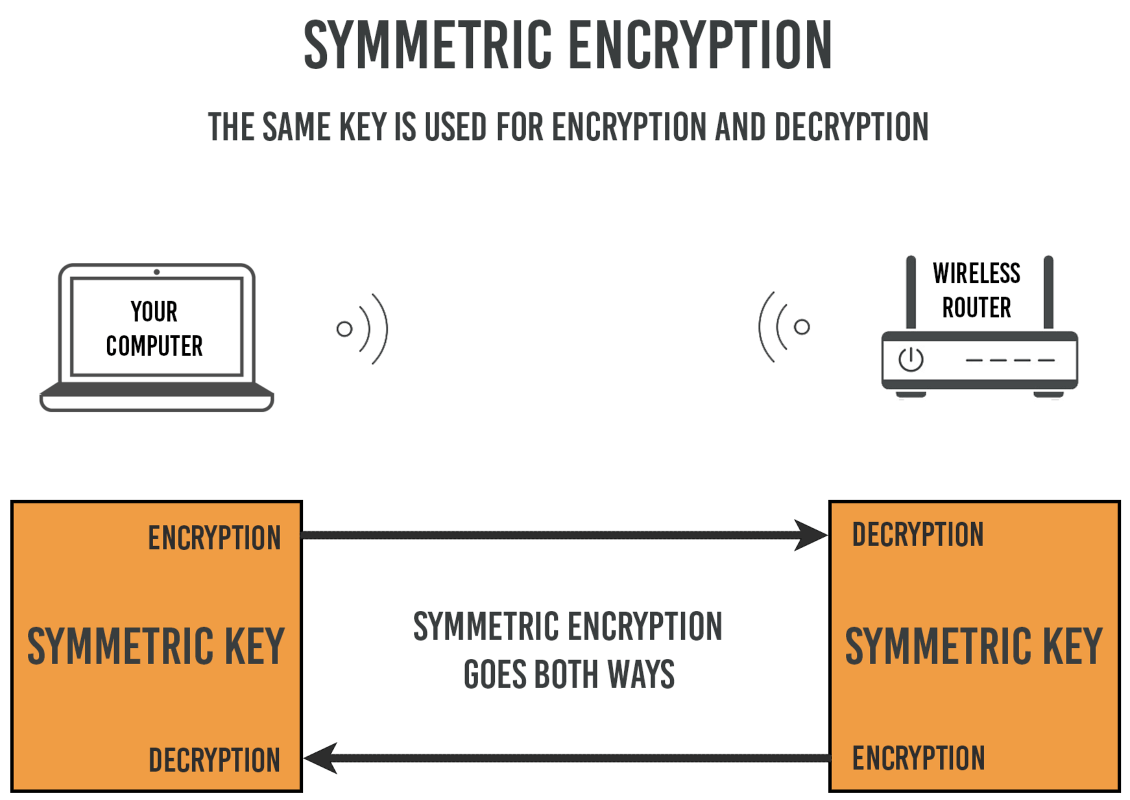
To *de*crypt the cipher text on the other end, you just reverse the process with the same key and it reverses the encryption, restoring the original form of the data. It's the secrecy of the encryption key that makes the whole process work. Only the intended recipients of the data should have it, or else the purpose is defeated.

When you use the same encryption key on both ends it's called *symmetric* encryption. This is what your home WiFi uses. You have just one key, or "password", which you plug into both your wireless router and your laptop. Easy peasy.



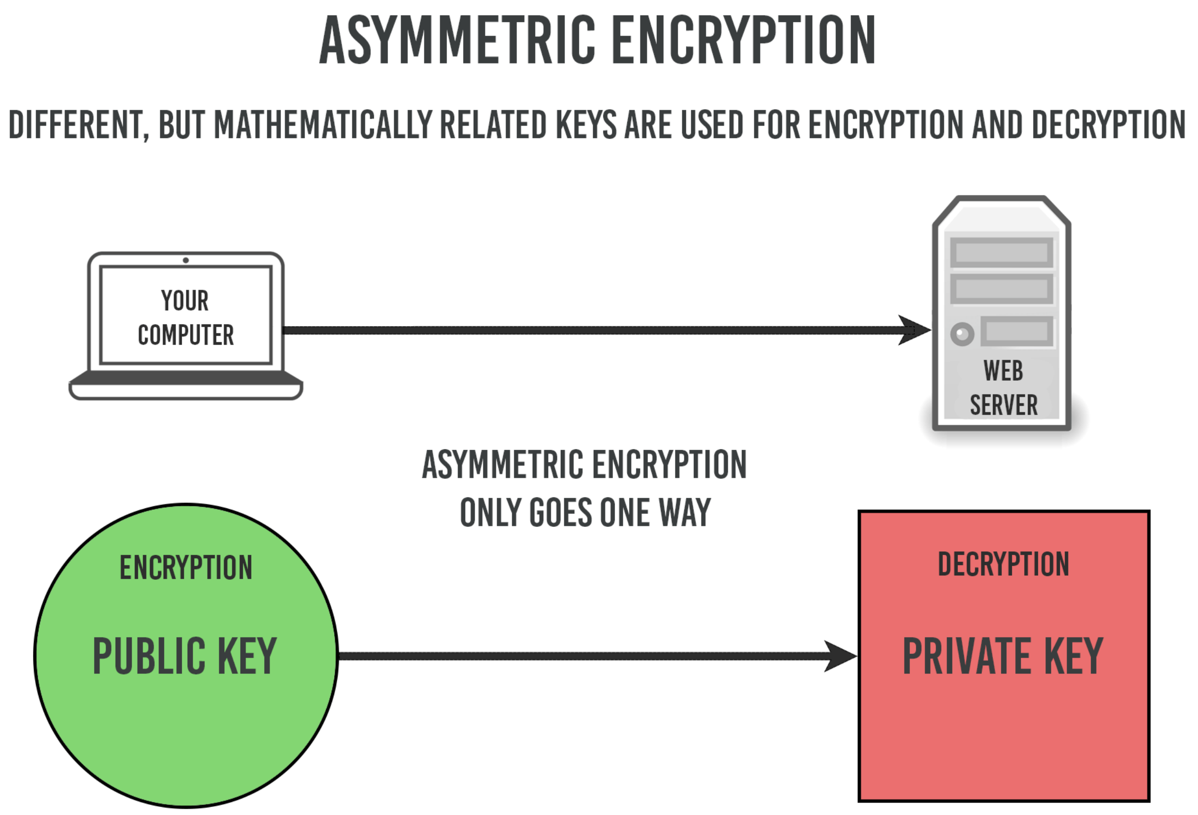
What to Do Before Taking Your Computer to the Repair Shop

ADVERTISEMENT



But it becomes more complicated when connecting to a website on the public internet. Symmetric encryption, by itself, won't work because you don't control the other end of the connection. How do you share a secret key with each other without the risk of someone on the internet intercepting it in the middle?

This problem is solved with *asymmetric* encryption. Asymmetric means you're using two different keys, one to encrypt and one to decrypt. We also call this *Public Key Cryptography* because it's how we establish secure connections on the public internet.


ASYMMETRIC ENCRYPTION
DIFFERENT, BUT MATHEMATICALLY RELATED KEYS ARE USED FOR ENCRYPTION AND DECRYPTION
YOUR COMPUTER
WEB SERVER
ASYMMETRIC ENCRYPTION ONLY GOES ONE WAY
ENCRYPTION
PUBLIC KEY
DECRYPTION
PRIVATE KEY

Key-pairs

To understand asymmetric encryption, you need to know how two separate keys can encrypt and decrypt the same data. As it turns out, it's just a math problem with *very* large numbers.

It requires a special mathematical process using very large prime numbers and modular arithmetic, among other things. The technical details are beyond the scope of this article but this is how it works conceptually.

Typically (not always) both the public and private keys are computed together at the same time, in the same mathematical process. This means they're *strongly related*, mathematically speaking. Because of this relationship, they can be used to encrypt/decrypt the same data. And that is also why public and private keys from different key-pairs would not work together. Every web server has its own unique set, making your connection to the website unique from other sites.

However, the process can only go one direction. When one of the keys (either public or private) is used to *encrypt* some data, only the other key can be used to *decrypt* it. That's just how the math conveniently works.

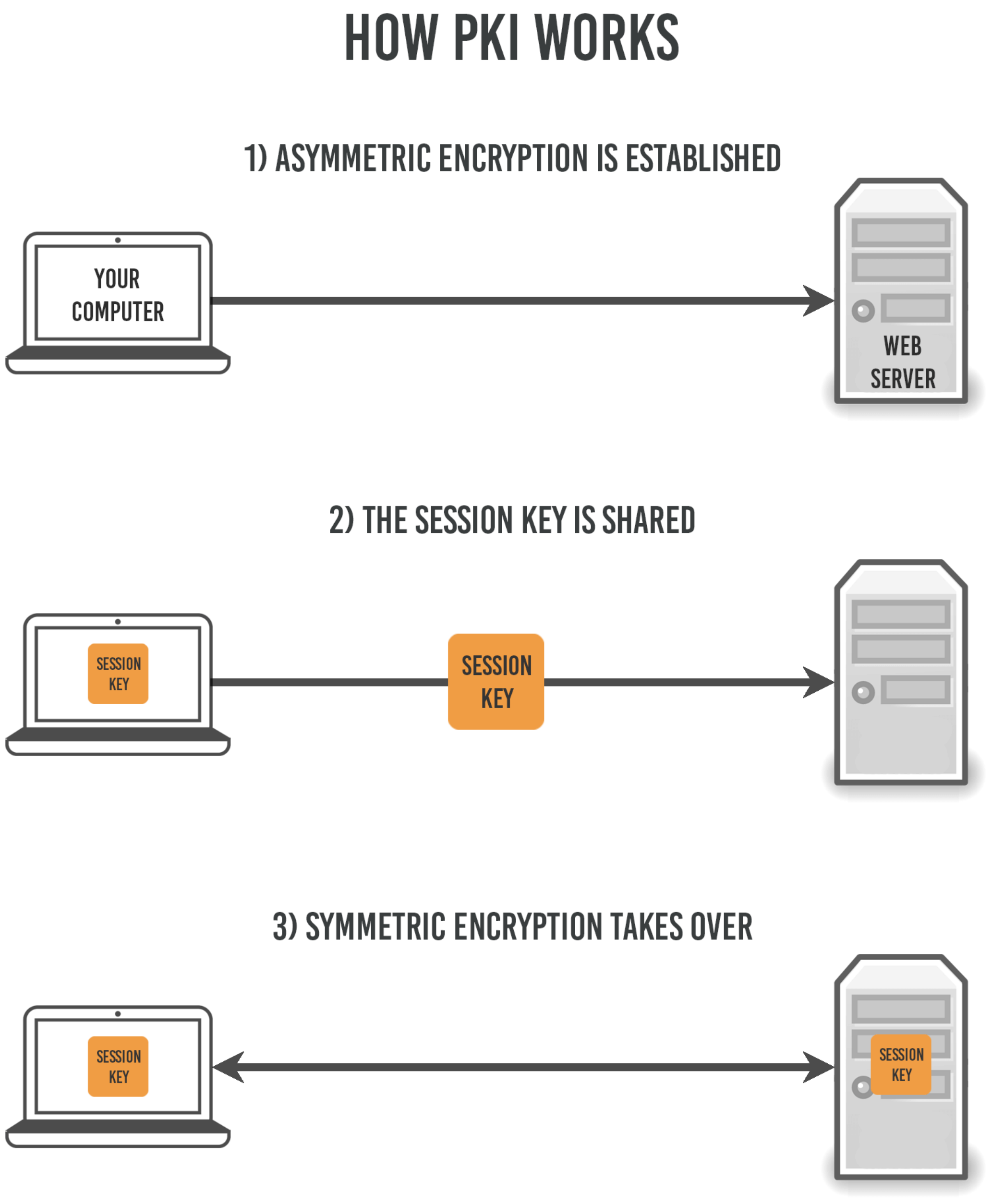
So it doesn't matter who else has the public key because it's worthless once the data has been encrypted. It can only be decrypted with the private key, which is stored in secret on the web server.

More about key-pairs:

When very large prime numbers are multiplied together, they're essentially impossible to factor ("unmultiply") without knowing what the original numbers were. It's not magic, it just happens to be the way the math works with prime numbers. In order to crack the encryption, you would have to be able to factor the product of the multiplied primes. It's technically possible that someone will figure out how to do this some day, but based on our current computing power, the foreseeable future seems safe. At least until quantum computing comes of age.

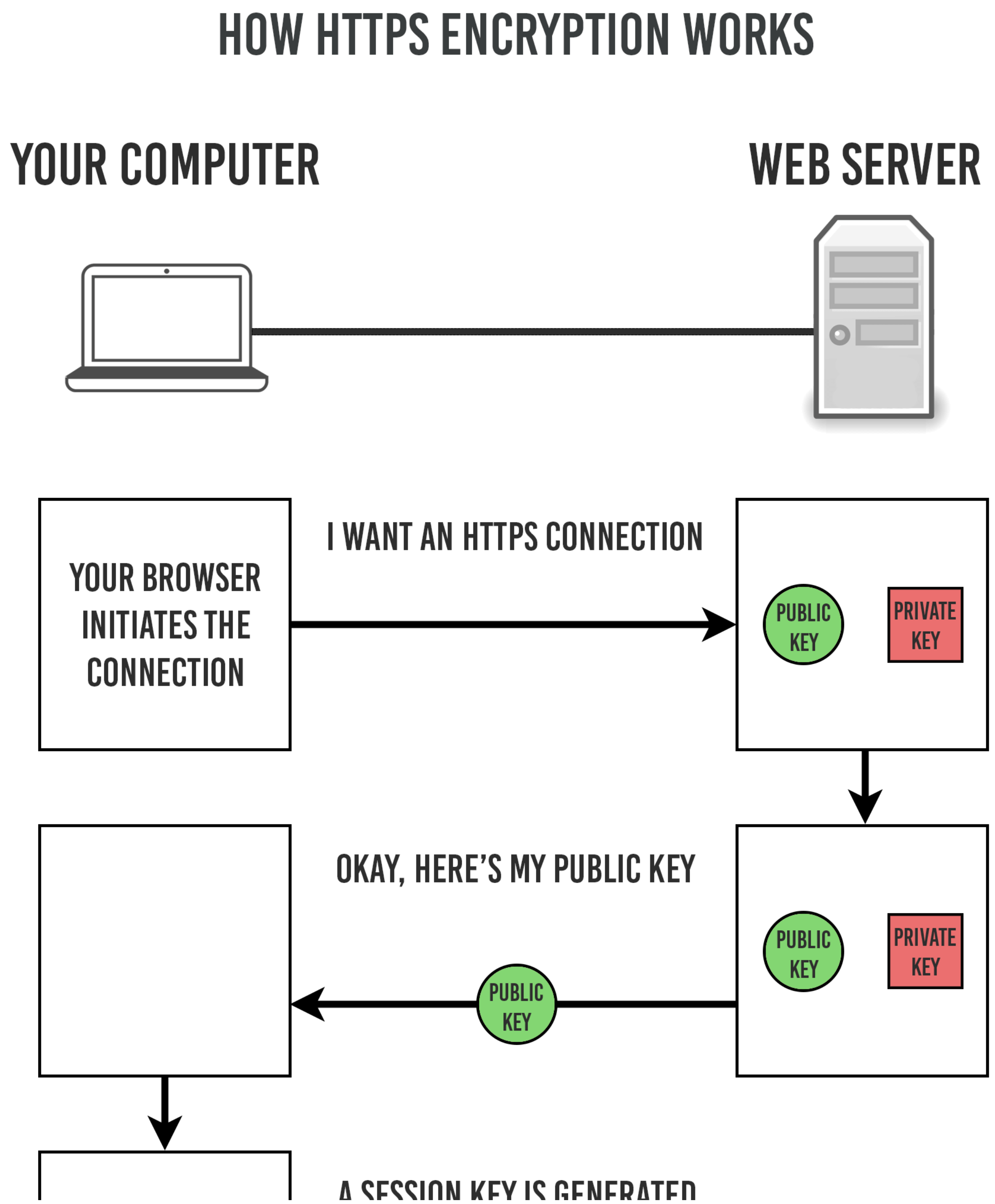
How Public Key Cryptography Works

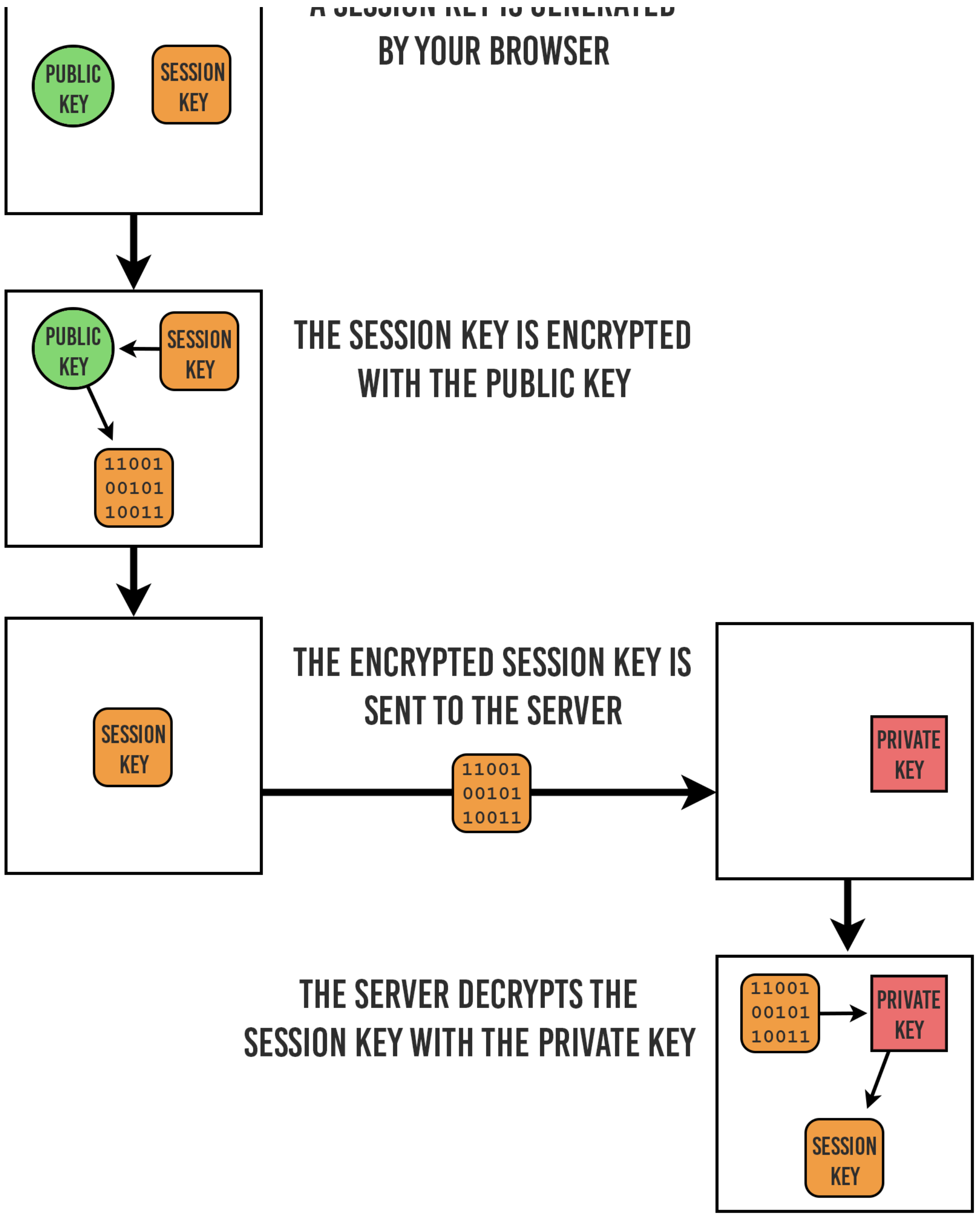
In the Public Key Infrastructure (PKI), both types of encryption are used. Asymmetric (public key) encryption is used first to establish the connection, which is then replaced with symmetric encryption (called the *session*) for the duration.

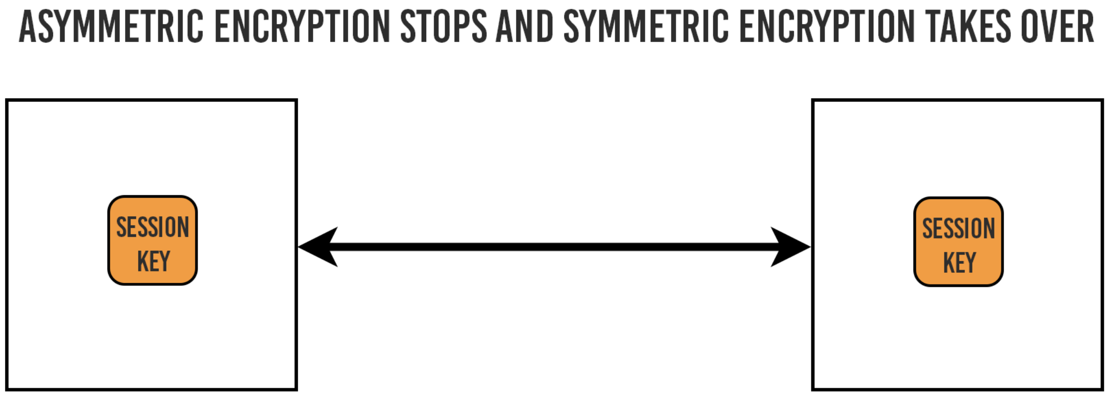

HOW PKI WORKS
1) ASYMMETRIC ENCRYPTION IS ESTABLISHED
YOUR COMPUTER
WEB SERVER
2) THE SESSION KEY IS SHARED
SESSION KEY
SESSION KEY
3) SYMMETRIC ENCRYPTION TAKES OVER
SESSION KEY
SESSION KEY

Here’s how it works in more detail:

1. Your browser reaches out to the website server and requests a connection.
2. The server sends you its public key. It keeps its private key a secret.
3. Your browser generates a third key called a *session key*.
4. The session key is encrypted by your computer using the public key you got from the server
5. The encrypted session key is then shared with the server.
6. The server decrypts the session key that it received from you using the secret private key. Now both ends have the session key that your computer generated.
7. The public key encryption is terminated and replaced with symmetric encryption.
8. Now you are in a session with the server using only symmetric encryption, and that’s how it remains until you leave the website.


HOW HTTPS ENCRYPTION WORKS
YOUR COMPUTER
WEB SERVER
YOUR BROWSER INITIATES THE CONNECTION
I WANT AN HTTPS CONNECTION
PUBLIC KEY
PRIVATE KEY
OKAY, HERE'S MY PUBLIC KEY
PUBLIC KEY
PRIVATE KEY
PUBLIC KEY
A SESSION KEY IS GENERATED


A SESSION KEY IS GENERATED
BY YOUR BROWSER
PUBLIC KEY
SESSION KEY
THE SESSION KEY IS ENCRYPTED
WITH THE PUBLIC KEY
11001 00101 10011
THE ENCRYPTED SESSION KEY IS
SENT TO THE SERVER
PRIVATE KEY
THE SERVER DECRYPTS THE
SESSION KEY WITH THE PRIVATE KEY


ASYMMETRIC ENCRYPTION STOPS AND SYMMETRIC ENCRYPTION TAKES OVER
SESSION KEY
SESSION KEY

As you can see, public key (asymmetric) encryption is only used briefly in the beginning to exchange the third key which is used for the rest of the connection. But what's the point of switching from asymmetric to symmetric? There are a couple reasons.

First, public key encryption only goes one way. Your encrypted data going to the website is secure only because the web server keeps the private key a secret. But if the server tried sending encrypted data back to you with the same key-pair, it would not be secure because *everyone* has access to its public key. That means anyone could decrypt it. You would have to establish two asymmetric sessions, one going each way. It's just not feasible for your computer to do that securely.

Second, the mathematical overhead for asymmetric encryption is far higher and therefore requires much more computing power to sustain. It is not suitable for long sessions because of the processing power it takes to keep it going. Public key encryption uses *much* longer keys, which makes it far more labor-intensive. Check out the key examples below.

Symmetric encryption key

AES 256-bit session key (expressed in hexadecimal)

```
C8D5897DCC56D6D462B8F32D464303161ACE11E536F04AE1
```

Asymmetric encryption key

RSA 2048-bit private key (expressed in base64)

```
MIIEogIBAAKCAQBl1esMuCZXUjwiBaUJsZlHrcGJ988fblnhcTjtpnaovHYp7IZW
0EMIkKMuF6tILPuJBMd2FoFHC2ZUVcmGrFabK/zRzrEb74djiH4l1gQHZDsQYybv
Qm6bJEsT5Cy+lwRCvXznEnWmQu62HX09CThtXUPNwfGLcAEFQLzgKPOcU4DZboY9
Tlm/Repe9w6h0KEzB7MHFIt446RPh9FalQwfrjfxGNHb+8V9BVlpRetXxd6aj1oo
WEsg5TqrZB8Cr17l/KXh+goqlyFmsz6WMpbv5oLbG+535PhyZjJ/VjiuTi6jsD9N
Skgq+iEbx74ZE855Ba7iYlJma6vxAj3ILdyLAgMBAAECggEAB41BjQPq8/bJNsys
XHerIkGkZJLX7UDFsY4v5o+9pO205Y8At1diYTQ4paZjsUqErLiKAhvRIm+Z+w8R
jscg9QjiCr0FLUkqBRuOH0grrFCmgKSZsox+n0qltqfpGYkha8GZjZk02EuGYEL3
kpsocBPGf2+udeSsbrNKTmU7C5CdOp+Fzmw4goltgmS3Sn8FWBaWOgblcUIFyupB
ETAzxUiH2qQom3plosMb11NzVNw8LajyGNmphB5szRHx+6Y6fGuhEacIePez8LJE
zRUdcjomzNNsft1iAduc0cZSRfPCMMmP989l+tWck2d/i9QWtuRoDhxsXyAaCieZ
dDnAIQKBgQDGrxF1WRum9N+D4D2MRTntseiNyZAKzgcPmXqyWoyPFDKsiTsk/Qe2
/KIQjJf11JcpZ+12UNNRFjghiFyl3Bh1E+pT6kzTPUZ5Ewws0hjm5CL667RkWCtn
4N4BoWPLmM5KFXMvmoK3iNyMLI+LZmI3HQ2wLcpC2+I1cpVmyQHoZQKBgQCDNotl
s2tuzNOKUsgrYl3SGk3MgOHe147h/Q32/x+dahIihVz6N3/nCiVpmliySGuLlxOu
OsWJby7XNuana7AVt59pE7G87pHtNFpLvhmN8oBDgvzbH2pLta/fQM9yjDokyk0I
4jhn2cMDE6qpx1yLnR8Hogb393GXqmAB/RtKLwKBgCF1yE3n9NU4NhBCinr+CZRz
OsPuJxj6u4uwAE7bGwLZlvMhpVlEV7FEeJKq6siWxeJOQ1qCoCPuxwM2sMup9mB5
OyRouAl3L1AimhFOTK1NzGpPmbCwfJbg8uJ5aJIYKN2nIR/qOItqPCJLOt1yH6ZC
eVSrUX3TBBSWOHz36x9VAoGAPyB8zD8XzsTIeWH+X7EBhlWVXHAdOwLmAR+oGk7/
vNINS4JyCwesh1FAUqApUAy7gPr8QvSPmOFe3bfSHHYYzzn6fak197mnh0GKt6oL
zVkAEHryf9GLaEVPCc+6flesmYqiqUV88am4wsAxeWEqyxialyeCxsXLZrZXXl2R
RY0CgYEAioZF15++gWnRzVKL/xFMj6Si53hekXHMrAsyYkxDB7Drjz3Qg9QN7fG1
gzBXLeRPa0tnkfcWgeXIdbHNZywfqrhaUO+k4t5Ei5n8ZXDKeqtNAGSrVdG282/a
OXL64xujKmZG27kZjK43bRe27obUHXbV7X0unT98QyYyWmkXdKk=
```

Symmetric encryption keys can be much shorter because no part of them are ever made public. It’s the public nature of asymmetric encryption that requires the long keys. Because when you have the public key, you already have part of the answer. Computing the rest of the answer (the private key) would be easy if the answer was short. Having exponentially larger keys is what makes it feasible to keep the private key private.

What HTTPS Does Not Do

It's easy to think of HTTPS as a miracle security solution for the internet, but there is a lot that it can't do.

HTTPS does not:

Hide the *names* of websites that you're visiting

This is because the name (aka "domain") of the website is sent using DNS (domain name service), which is not inside the HTTPS tunnel. It is sent before the secure connection is made. An eavesdropper in the middle can see the name of the website you're going to (e.g. TipTopSecurity.com), they just can't read any of the actual content that's being transferred back and forth. It won't be until DNSSEC is fully implemented that this will change.

Protect you from visiting an evil website

HTTPS does not ensure that the website, itself, is safe. Just because you're connecting securely doesn't mean you're not connecting to a website run by bad guys. We try to fix this problem with trusted Certificate Authorities but the system isn't perfect (stay tuned for more on this).

Provide anonymity

HTTPS does not hide your physical location or personal identity. Your personal IP address (your address on the internet) has to be attached to the outside of the encrypted data, because the internet wouldn't know where to send it if your IP address was encrypted, too. And it also doesn't obscure your identity to the website you're visiting. The site you visit still knows everything about you that it would on a non-secure connection.

Prevent you from getting viruses

HTTPS is not a filter. It's possible to receive viruses and other malware over an HTTPS connection. If the web server is infected or you're on a malicious website that's handing out malware, it will be sent inside the HTTPS stream just like everything else. HTTPS *does*, however, prevent anyone in the middle from injecting malware into your moving traffic.

Protect your computer from being hacked

HTTPS only protects the data while it's moving between your computer and the web server. It does not offer any protection for your actual computer or the server, themselves. This also means that if there's malware that's monitoring traffic on one end of the connection, it can read the traffic before and after it's encrypted inside the HTTPS stream.

Basically, HTTPS only protects your information while it's flowing through the wires (or the air). It can't protect your computer, your identity, or hide which sites you're visiting. HTTPS is only one part of a safer internet. If you're looking for more privacy then a VPN service would be the next step. Check out this article for more about VPNs.

Authentication

There's another side to secure websites that we haven't even begun to cover. It's one thing to create an HTTPS connection, but how can you actually trust the web server you're connected to? It doesn't matter if your connection is encrypted if you're connected to a phishing website that's trying to steal your info.

We chose to solve this problem by implementing a third party system of *Certificate Authorities*. I'll be covering this topic eventually, so if you'd like to know when the article goes live then be sure to sign up for my newsletter.