

Forum:[Hack Forums](#)

Thread: [Legion | \[TUT\] Base64 encoding with a pen and paper](#)

J?mes - March 15, 2013, 11:03 pm

[img]http://i.minus.com/ieq6viOQ6szdE.gif[/img]

I thought I might make a thread on how to base64 encode and decode to give everyone a bit more of an insight into basic cryptography. If you don't understand any part, please post here and I'll fix it up. This tutorial will require you to read every single step to understand it. You will also need two images which you can print or just show on your screen for this. For someone who is advanced in cryptography, I may use really basic terminology, but it's so that people can understand it. For this tutorial, we will base64 encode the word "Hello" into "SGVsbG8=".

Step 1

Get out a piece of paper and a pen and print or display the following 2 image on your screen

Spoiler ([Click to View](#))The ASCII character table

[img]http://www.asciitable.com/index/asciifull.gif[/IMG]

Spoiler ([Click to View](#))The base64 character table

Step 2

Draw the following boxes up on your piece of paper

Spoiler ([Click to View](#))

Step 3

Now here is where it all starts. Firstly, you will need to understand thatAll characters on the ASCII table (all special, numeric, and alphabetical) characters have a specific number value on the ACSII table.

Binary only consists of 1s and 0s. To make our decimal value on the ACSII table actually become binary, we must use a binary table (first table you drew).

Each alphabetical character represents 8 binary numbers (8 zeros and ones).

All alphabetic characters are case sensitive which means that you need to assign the right number to the right case too. E.g. a on the ASCII table is 97 decimal value and A on the ASCII table is 65 decimal value.

So what do we start off with in this step? We firstly need to turn our characters into decimal values. These decimal values are on the ACSII chart as 'dec'. So we need to find the decimal value for each of our letters in Hello.

Code H = 72

e = 101

l = 108

l = 108

o = 111 That part is probably the easiest part in this step, but you will begin to understand everything soon enough. So we can now fill in the first two columns of our chart.

So what's all those random numbers for the rest of our columns? As I said before, each character will represent a 8 bit / character binary sequence. For example, 'H' has the decimal value of 72 and the binary value of 01001000. How is this possible though? For any character binary blocks, each number represents how much of another number is inside that number.

Binary goes up in 2s, so you will see a binary table of 128 64 32 16 8 4 2 1. The more characters in a binary block, the more x2 we go up to. For example, having a 10 character binary block, you would have 512 256 128 64 32 16 8 4 2 1.

In our case, the ASCII table only has a max value of 255 characters, which a 8 bit block of 128 64 32 16 8 4 2 1 can cover. Solving the binary value for H is pretty straight forward. You have the decimal of 72, and you go through the following process:

Code H = 72

How many 128's in 72? 0

We still have the value of 72

How many 64's in 72? 1

Now our remainder value is 8

How many 32's in 8? 0

How many 16's in 8? 0

How many 8's in 8? 1

Now we have nothing left. So..

How many 4's in 0? 0

How many 2's in 0? 0

How many 1's in 0? 0

What do we end up with? 01001000 which is our binary value for H or 72.

So, lets fill in our binary values for H, e, l, l, o.

If you still don't understand by this point, make sure you post and I'll try help you out.

Step 4

This is where we go from our binary representations of our characters to the base64 representations of our characters. So we currently have

Code H = 01001000

e = 01100101

l = 01101100

l = 01101100

o = 01101111 But how do we turn this into SGVsbG8=? The first task we have to do is put all of our 8 character binary blocks together. So we get the value

Code 0100100001100101011011000110110001101111 This is the point where we switch to base64 mode. Instead of re-splitting our blocks up into 8 characters, we split them up into 6. So lets see what we get...

Code 010010

000110

010101

101100

011011

000110
1111

BUT WAIT A SECOND. We have one block of 6 that only has 4 characters. This is because we have a total of 40 binary numbers BUT 40/6 isn't actually a full number. So what we are going to do is add two added zeros on the end. We need to remember they are two added 0's because we will have to add a equals sign in the padding step per 2 0's we have. In this case, we only have one pair of two zeros, but we will get that sorted later. So lets add the added zeros on the end to make our blocks all consist of 6 characters.

Code 010010
000110
010101
101100
011011
000110
1111(00) <---- added 0's.

Now, we are onto our second table. Our bin number is just each block. In this, you can see that we only have 32 16 8 4 2 1 instead of 128 64 32 16 8 4 2 1. This is because we went from 8 bit / character binary blocks to 6 bit / character binary blocks. So, we pretty much just do the reverse of what we did for the first step and insert each binary number into its respective column. For example, the first block of 010010

Code 0 of 32's
1 of 16's
0 of 8's
0 of 4's
1 of 2's

0 of 1'sWe are going to add the total of that column up and put it into our total column. So the total of that is $16+2 = 18$. If we do that for all of our blocks of 6, we should get something on our table looking like

So we have our base64 numerical values (18, 6, 21, 44, 27, 6, and 60) and we just need to use our base64 chart to find the character to the number.

Code 18 = S
6 = G
21 = V
44 = s
27 = b
6 = G
60 = 8

So our final value would be SGVsbG8 but we are not actually finished. Remember the added 0's? Go to step 5 if you end up with added 0's.

Step 5

This is the last step and you only need padding if you end up having those added zeros on the end. Per pair of added zeros, we add one equals sign onto the end of it. It's only possible to either have zero added 0's, two added 0's, or four added 0's. So in saying that, it's only possible to have either no equals, one equals or two equals. In our case, we had 2 added 0's, so we only add one equals sign to the end of it.

Our final value is SGVsbG8= . Want to check if this is correct? Visit <http://www.base64decode.org/> and enter in that value and see if this is correct :). Decoding base64 is pretty much the opposite process of what we just did. For example Find the numerical values for the base64 characters

Put them into binary (6 character blocks).

Put them into a long string

If you had one or two equals signs take off 2, or 4 0's respectively.

Put them into 8 character blocks

8 character blocks to numerical number

Lastly, numerical number to ascii character.

If you didn't understand the above explanation, tell me and I'll make an example of reversing what we just did. This thread seems a little silly but it was more about learning the basics of encoding than learning how to write down a base64 encoded message with a pen and paper.

Daring - March 15, 2013, 11:05 pm

Sorry, but when would you ever realistically need to use a pen and paper?

blue++ - March 15, 2013, 11:07 pm

(Mar 15, 2013 11:05 PM)Daring Wrote: [\[CITE\]](#) Sorry, but when would you ever realistically need to use a pen and paper?

When you're stuck in Iraq in a cold, wet cell and need to get a message out.

JK, I don't see it relevant either - but, it is a lengthy and in-depth tutorial. Give the guy some credit.

J?mes - March 15, 2013, 11:09 pm

(Mar 15, 2013 11:05 PM)Daring Wrote: [\[CITE\]](#) Sorry, but when would you ever realistically need to use a pen and paper?

It's not so much about actually base64 encoding / decoding with a pen and paper as it is understanding how it all works. This is what I'm trying to do here. Most people would just go on a website, click decode and be like, "YEP, DONE, SWEET" but some people actually want to know how it works which is what this thread is about.

Some skills in this thread are vital skills in understanding parts of Cryptography, that's why I wrote it up ;).

Debroski - March 15, 2013, 11:27 pm

(Mar 15, 2013 11:09 PM)J?mes Wrote: [\[CITE\]](#) It's not so much about actually base64 encoding / decoding with a pen and paper as it is understanding how it all works. This is what I'm trying to do here. Most people would just go on a website, click decode and be like, "YEP, DONE, SWEET" but some people actually want to know how it works which is what this thread is about.

Some skills in this thread are vital skills in understanding parts of Cryptography, that's why I wrote it up ;).

I agree, this also leads into the idea of how to make your own type as a lot of people seem to have a hard time grasping how to do that. This is a very well made tutorial, was a fun read and thanks for

contributing

J?mes - March 16, 2013, 4:31 pm

(Mar 15, 2013 11:27 PM)___Angel Wrote: [\[CITE\]](#) I agree, this also leads into the idea of how to make your own type as a lot of people seem to have a hard time grasping how to do that. This is a very well made tutorial, was a fun read and thanks for contributing

Thanks, at least someone understands :). I'll try make it more colourfull and appealing so more people actually read it :P.

XOwnage - March 16, 2013, 4:32 pm

Wow finally there is something besides crypters in the cryptography section good job. And even again wow it's not true crypt based :)

Keep it up really informative thread

Vinnie - March 17, 2013, 8:15 am

(Mar 16, 2013 04:32 PM)XOwnage Wrote: [\[CITE\]](#) Wow finally there is something besides crypters in the cryptography section good job. And even again wow it's not true crypt based :)

Keep it up really informative thread

Yes, James is pretty good at what he does. One of my favorite members in Legion.

Freerunning - March 17, 2013, 8:24 am

I love your tutorial seriously. I always wanted to learn cryptography

Anarchy - March 17, 2013, 8:26 am

Great tutorial James, hope to see you make more.

bannable - March 23, 2013, 9:54 pm

Just wanted to say thank you for your time and effort you put into this. It was a good read. Hope other find it helpful too.

CYB3RB0X - August 19, 2013, 8:25 pm

Really nice tutorial well made and easy to understand :)

TheDubWubCub - August 19, 2013, 9:05 pm

Awesome tutorial! Have been looking for this forever! Thanks! :D

chippydog - August 20, 2013, 1:48 am

Thanks for the tutorial buddy, awesome tutorial, definitely easy to understand and well written. HQ

hax unit - August 20, 2013, 8:04 pm

Sad to see your account closed. But this was really easy to understand and really cool to learn. Thanks

zagazeta - September 11, 2013, 11:42 am

Great tut! An easier way to learn base64 than how I was taught.

?XilluX - September 11, 2013, 11:44 am

Look, I have already figured this out my self and when you want to encode a message with 50 chars, will take you almost hours.

Nezekan - September 11, 2013, 12:08 pm

(Mar 15, 2013 11:05 PM)Daring Wrote: [\[CITE\]](#) Sorry, but when would you ever realistically need to use a pen and paper?
doomsday

when the robots attack

or something like that

V3gapUnk - December 17, 2013, 6:02 pm

Very Nice TUT!!thanks for this !!! (UNDESTOOD EVERYTHING)

Imperious1 - January 2, 2014, 12:07 pm

Looks nice, i'll try it out later when I have some time

DClive - January 8, 2014, 10:14 pm

thanks, I was looking for information and this is a great starting point for me

nepHaXOr - May 22, 2015, 8:59 am

Great tuts :) Thanks for this post

iLogic™ - May 26, 2015, 7:55 am

Thank you for this tutorial ? now i can decode base64 without computer/mobile ?