

RIDDLE 1

Riddle 1

Congrats, you have solved the first riddle, Your first key is: 6skd8s

GoogleForms

This content is neither created nor endorsed by Google.

Cryptii [Help us build the next cryptii](#)

VIEW

Ciphertext ▾

ozcjmz

1

ENCODE DECODE

Caesar cipher ▾

SHIFT

— 8 a→i +

ALPHABET

abcdefghijklmnopqrstuvwxyz

CASE STRATEGY

Maintain case ▾

FOREIGN CHARS

Include Ignore

→ Decoded 6 chars

VIEW

Plaintext ▾

gruber

RIDDLE 2

RIDDLE 2

Congrats for solving the second riddle, the key is: cy8snd2

[Submit another response](#)

Google Forms This content is neither created nor endorsed by Google.

01000111 01100101 01101110 01101110 01100101 01110010 01101111



Character encoding (optional)

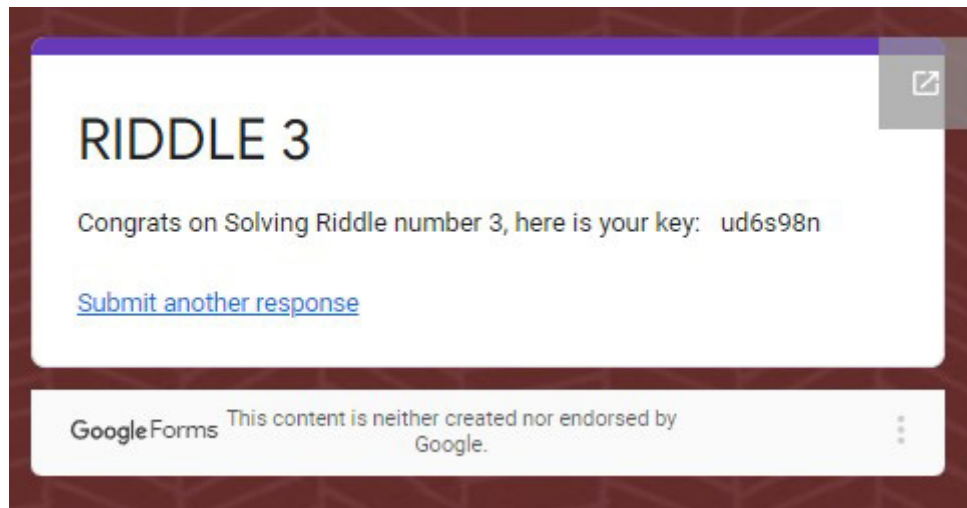
ASCII/UTF-8

↻ Convert

✕ Reset

↕ Swap

Gennero



```

sysadmin@ubuntudesktop: ~/Desktop
File Edit View Search Terminal Help

< You are always busy. >
-----
\
 \
  .--.
  |o_o|
  |:~/|
 //   \ \
(|     |)
/\'   \\'
 \___/  =  \___/

sysadmin@ubuntudesktop:~$ ls
Cybersecurity-Lesson-Plans  Documents  Music      Public  rockyou.txt  Videos
Desktop                     Downloads  Pictures   python  Templates

sysadmin@ubuntudesktop:~$ cd Desktop/
sysadmin@ubuntudesktop:~/Desktop$ touch riddle3.txt
sysadmin@ubuntudesktop:~/Desktop$ ls
image.jpg  riddle3.txt  sample-image.jpg  script.php
sysadmin@ubuntudesktop:~/Desktop$ nano riddle3.txt
sysadmin@ubuntudesktop:~/Desktop$ cat riddle3.txt
4qM0IvwEGXzvKmvRE2bNbg==
sysadmin@ubuntudesktop:~/Desktop$ openssl
OpenSSL> ^C
sysadmin@ubuntudesktop:~/Desktop$ openssl enc -pbkdf2 -nosalt -aes-256-cbc -in
riddle3.txt -d -base64 -K 5284A3B154D99487D9D8D8508461A478C7BEB67081A64AD9A1514
7906E8E8564 -iv 1907C5E255F7FC9A6B47B0E789847AED
takagi
sysadmin@ubuntudesktop:~/Desktop$

```

RIDDLE 4

Congrats! The Key is: 7gsn3nd2

[Submit another response](#)

GoogleForms

This content is neither created nor endorsed by Google.

**Jack and Jill went up a Hill to
use their public Keys**

**Jack had 2, and Jill did too
to exchange their messages
with ease.**

**What would Jack use to send
an encrypted message to Jill?**

- ☐ Jack's Public Key
- ☐ Jack's Private Key
- ☒ Jill's Public Key
- ☐ Jill's Private Key

The image shows a Google Form titled "RIDDLE 4". At the top right, there is a share icon. Below the title, the user's email "frank.ymlin@gmail.com" is displayed with a "Switch account" link and a checkmark icon. A red asterisk indicates a required question. The question is "What would Jill use to to decrypt Jacks message? *". It has four radio button options: "Jack's Public Key", "Jack's Private Key", "Jill's Public Key", and "Jill's Private Key", with the last one selected. At the bottom, there are "Back" and "Next" buttons, and a "Clear form" link. The footer shows the Google Forms logo and a disclaimer: "This content is neither created nor endorsed by Google."

RIDDLE 4

frank.ymlin@gmail.com (not shared) [Switch account](#)

* Required

Part 2



What would Jill use to to decrypt Jacks message? *

- ☐ Jack's Public Key
- ☐ Jack's Private Key
- ☐ Jill's Public Key
- ☒ Jill's Private Key

[Back](#) [Next](#) [Clear form](#)

Google Forms This content is neither created nor endorsed by Google.

RIDDLE 4

 frank.ymlin@gmail.com (not shared) [Switch account](#)  Draft saved

* Required

Part 3

Jack and Jill invited Bob, Alice, Tim and Peter along to exchange some messages. How many keys would they all need for asymmetric vs symmetric encryption? *


- ☐ 12 Asymmetric and 30 Symmetric
- ☐ 10 Asymmetric and 15 Symmetric
- ☐ 6 Asymmetric and 15 Symmetric
- ☐ 15 Asymmetric and 12 Symmetric
- ☒ 12 Asymmetric and 15 Symmetric

[Back](#)



[Next](#)

[Clear form](#)

Google Forms This content is neither created nor endorsed by Google.



RIDDLE 4

 frank.ymlin@gmail.com (not shared) [Switch account](#)  Draft saved



* Required

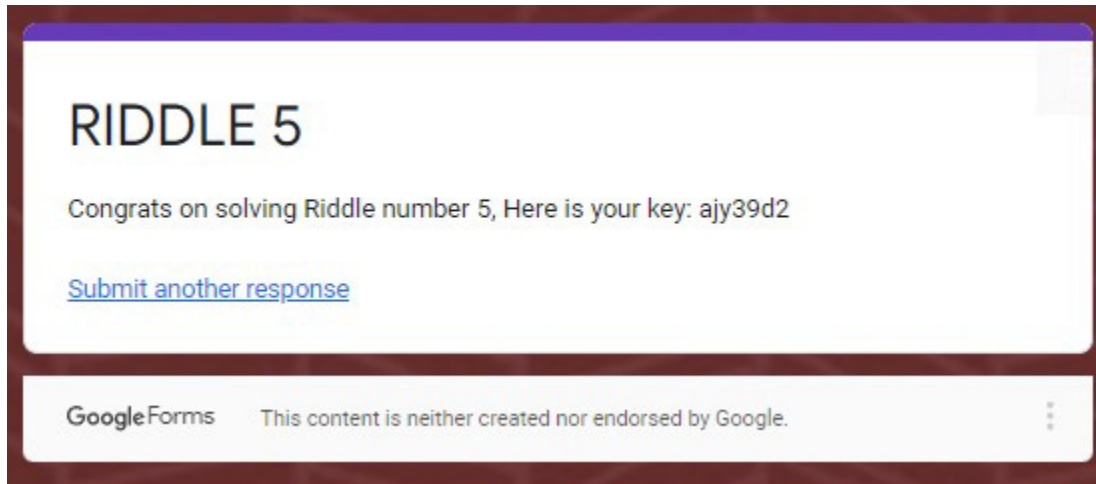
Part 4

Tim just sent an encrypted message to one of his friends, which of the following keys did he likely use to encrypt the message *

- ☒ Alice's Public Key
- ☐ Tim's Private Key
- ☐ Tim's Public Key
- ☐ Bob's Private Key
- ☐ Peter's Private Key

[Back](#) [Next](#) [Clear form](#)

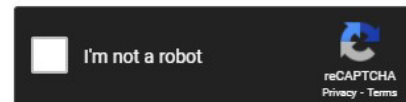
 This content is neither created nor endorsed by Google. 



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

3b75cdd826a16f5bba0076690f644dc7

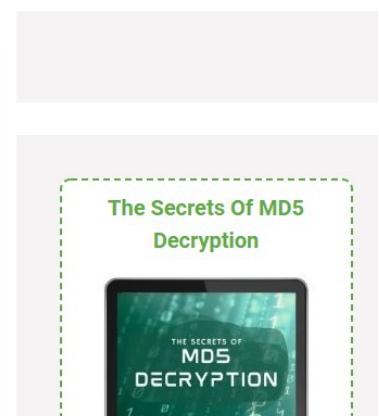
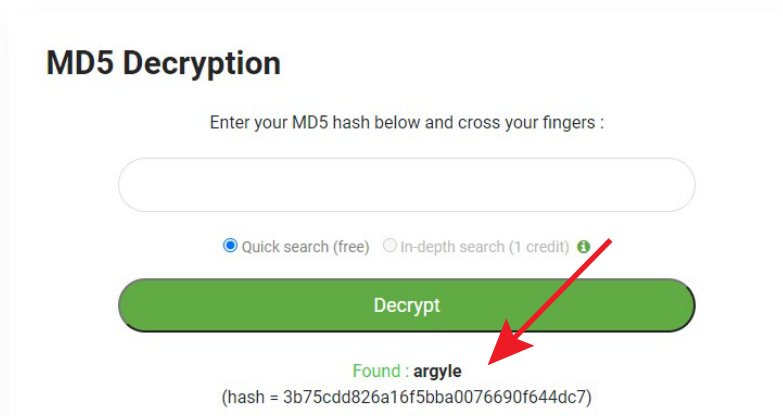


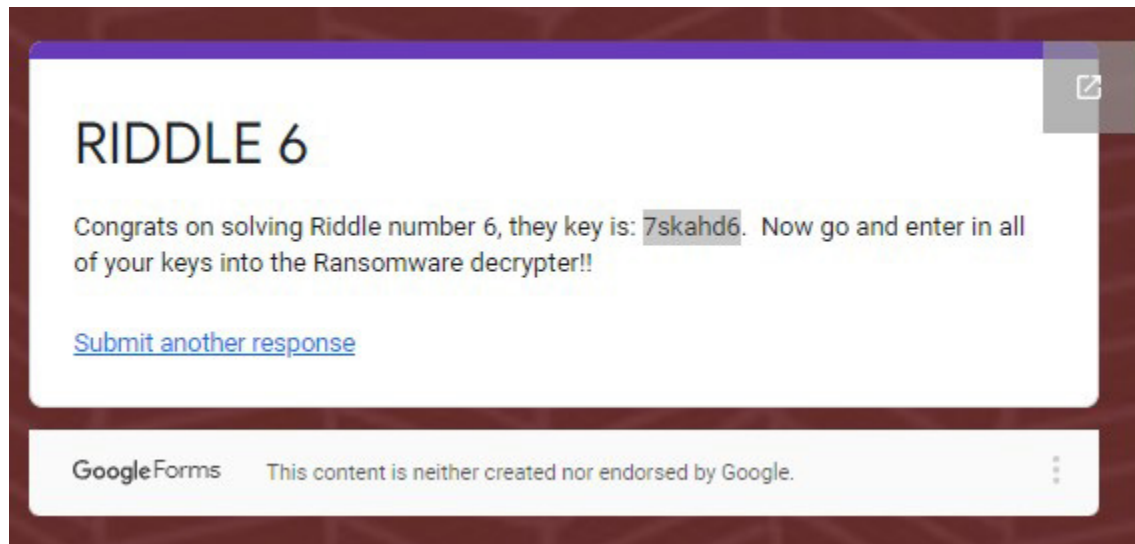
Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
3b75cdd826a16f5bba0076690f644dc7	md5	argyle

Color Codes: **Green** Exact match, **Yellow** Partial match, **Red** Not found.

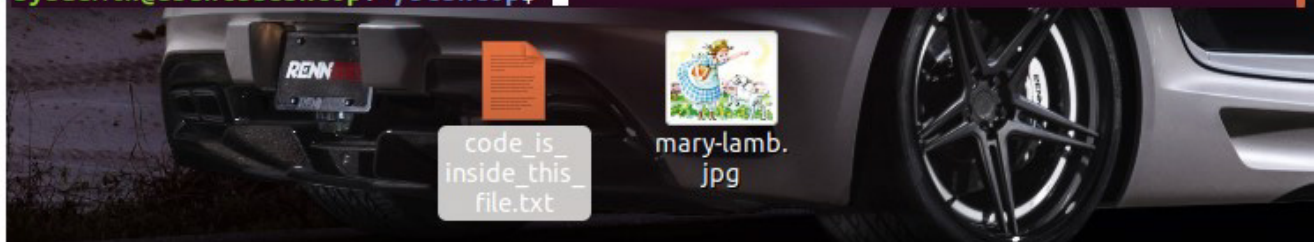




```
extracting options:
-sf, --stegofile      select stego file
-sf <filename>        extract data from <filename>
-p, --passphrase      specify passphrase
-p <passphrase>       use <passphrase> to extract data
-xf, --extractfile    select file name for extracted data
-xf <filename>        write the extracted data to <filename>
-f, --force           overwrite existing files
-q, --quiet           suppress information messages
-v, --verbose         display detailed information

options for the info command:
-p, --passphrase      specify passphrase
-p <passphrase>       use <passphrase> to get info about embedded data

To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
To extract embedded data from stg.jpg: steghide extract -sf stg.jpg
sysadmin@ubuntudesktop:~/Desktop$ steghide extract -sf mary-lamb.jpg
Enter passphrase:
wrote extracted data to "code_is_inside_this_file.txt".
sysadmin@ubuntudesktop:~/Desktop$ cat code_is_inside_this_file.txt
mcclane
sysadmin@ubuntudesktop:~/Desktop$
```





Please send your 100 Bitcoin to the following wallet anyways:

bc1q7ppkvhdqr0ml4v2fsda2lt85q7jm8yy7dytxet