

LABORATORY

CEL53.2: Cryptography and System Security Winter 2021-22

Experiment 7:	Creating and Reading QR Codes Ref: https://www.linux-magazine.com/Online/Features/Generating-QR- Codes-in-Linux
--------------------------	---

Name: Ameya Jangam

UID: 2019130025

Branch: TE Comps

Note: Students are advised to read through this lab sheet before doing an experiment. On-the-spot evaluation may be carried out during or at the end of the experiment. Your performance, teamwork/Personal effort, and learning attitude will count towards the marks.

Experiment 7: Creating and Reading QR codes

1 OBJECTIVE

Creating and Reading QR Codes for given exercise

2 INTRODUCTION

With the right tools, you can create your own QR code squares with information you want to share on a business card, in a letter, or on your website.

Read errors of unreliable, one-dimensional bar codes often caused interruptions in industrial production, prompting companies like Toyota and its subsidiary Denso Wave to develop as early as 1994 a new code for acquiring stock data. The new matrix bar code was designed to store more information than the traditional bar code and to stay legible, even if the label was dirty, wrinkled, or partially destroyed.

The quick response code, or QR code, comprises a matrix of square dots instead of the usual lines. Measuring up to 177 by 177 dots, the QR code encodes up to 4,296 characters, compared with a barcode that encodes just 13.

Thanks to numerous free reader apps for smartphones, QR codes have gained in popularity in recent years. Posters, catalogs, magazines, business cards, and even television screens display the small squares, offering additional information or URLs for microsites.

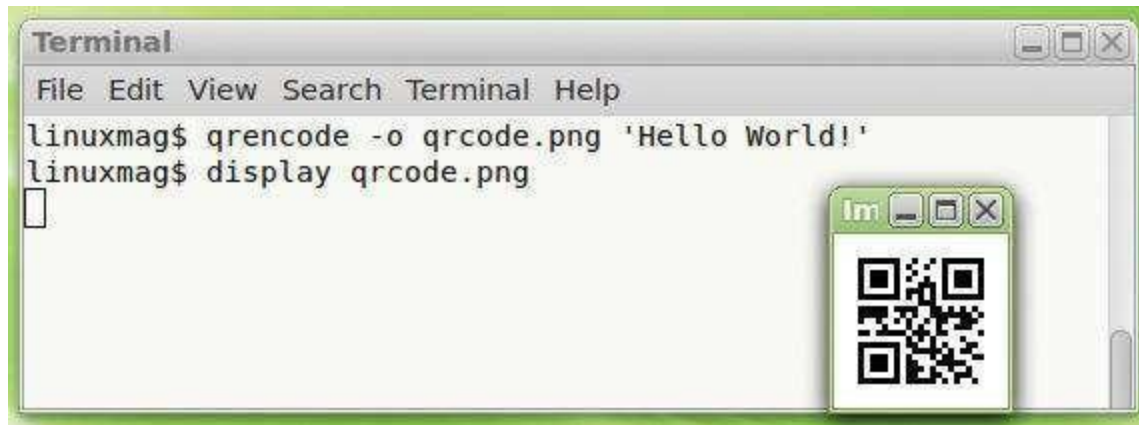
Data Grabber

If you want to encode your address into a QR code for a business card, you could try one of the many services on the Internet. However, restrictions typically apply. For example, some services allow only QR codes in a certain size or in limited quantities. Also, you never know for sure, what the services do with your data. For example, some sites explicitly allow sharing and selling the data to third parties in their terms of use, more or less pre-programming an increase in junk mail in the future. Fortunately, several QR code tools can help you avoid these problems in Linux.

3 PROCEDURE/TASK

The quickest way to create a QR code is with the [Qrencode](#) command-line utility. Any major distribution can install Qrencode via the package manager. The following command then creates a QR code containing the text “Hello World!”:

```
$ qrencode -o qrcode.png 'Hello World!'
```



QR Code relevant Hello World will pop up: Qrencode encoding the string “Hello World!” as a QR code and storing it in the qrcode.png file.

The generated QR code ends up in the *qrcode.png* file. If the file already exists, it’s overwritten without prompting. If you have at least version 3.3.0 (*qrencode -V*), Qrencode can generate an EPS graphic,

\$ *qrencode -t EPS -o qrcode.eps 'Hello World!'* or

ASCII output.

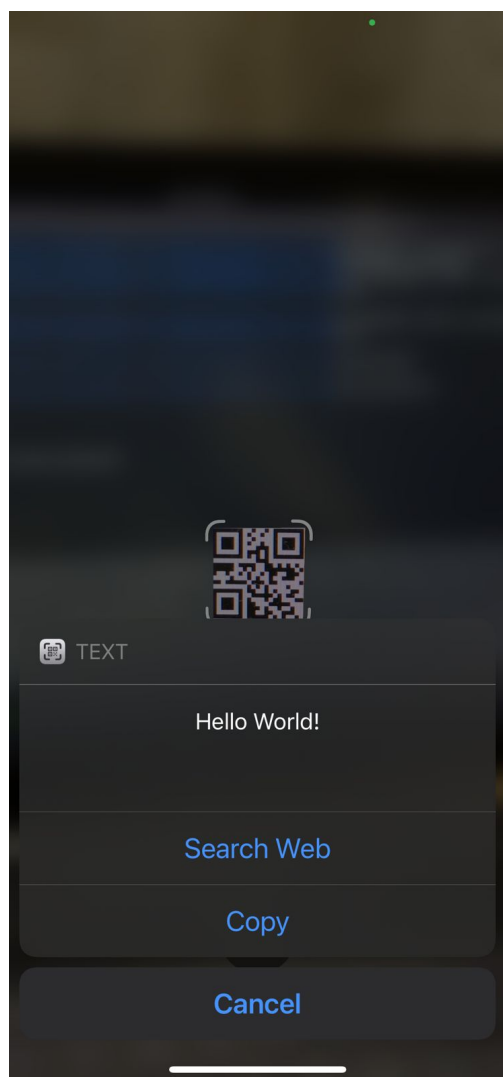
```
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7$ qrencode -t EPS -o qrcode.eps 'Hello World!'
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7$ ls
qrcode.eps  qrcode.png
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7$ qrencode -t EPS -o qrcode.txt 'Hello World!'
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7$ ls
qrcode.eps  qrcode.png  qrcode.txt
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7$ cat qrcode.
cat: qrcode.: No such file or directory
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7$ cat qrcode.txt
%!PS-Adobe-2.0 EPSF-1.2
%BoundingBox: 0 0 87 87
%Pages: 1 1
```

```

ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7$ cat qrcode.txt
%!PS-Adobe-2.0 EPSF-1.2
%%BoundingBox: 0 0 87 87
%%Pages: 1 1
%%EndComments
/p { moveto 0 1 rlineto 1 0 rlineto 0 -1 rlineto fill } bind def
gsave
1.000000 1.000000 1.000000 setrgbcolor
87 87 scale
0 0 p
grestore
0.000000 0.000000 0.000000 setrgbcolor
3 3 scale
4 24 p 5 24 p 6 24 p 7 24 p 8 24 p 9 24 p 10 24 p 13 24 p 16 24 p 18 24 p 19 24 p 20 24 p 21 24 p 22 24 p 23 24 p 24 24 p 4 23 p 10 23 p 12 23 p 15 23 p
4 24 p 5 24 p 6 24 p 7 24 p 8 24 p 9 24 p 10 24 p 13 24 p 16 24 p 18 24 p 19 24 p 20 24 p 21 24 p 22 24 p 23 24 p 24 24 p 4 23 p 10 23 p 12 23 p 15 23 p
18 23 p 24 23 p 4 22 p 6 22 p 7 22 p 8 22 p 10 22 p 13 22 p 18 22 p 20 22 p 21 22 p 22 22 p 24 22 p 4 21 p 6 21 p 7 21 p 8 21 p 10 21 p 12 21 p 15 21 p
18 21 p 20 21 p 21 21 p 22 21 p 24 21 p 4 20 p 6 20 p 7 20 p 8 20 p 10 20 p 14 20 p 15 20 p 16 20 p 18 20 p 20 20 p 21 20 p 22 20 p 24 20 p 4 19 p 10 1
9 p 12 19 p 13 19 p 14 19 p 16 19 p 18 19 p 24 19 p 4 18 p 5 18 p 6 18 p 7 18 p 8 18 p 9 18 p 10 18 p 12 18 p 14 18 p 16 18 p 18 18 p 19 18 p 20 18 p 21
18 p 22 18 p 23 18 p 24 18 p 14 17 p 15 17 p 16 17 p 4 16 p 5 16 p 6 16 p 7 16 p 8 16 p 10 16 p 11 16 p 12 16 p 13 16 p 16 16 p 17 16 p 19 16 p 21 16 p
23 16 p 4 15 p 5 15 p 6 15 p 7 15 p 8 15 p 11 15 p 13 15 p 16 15 p 17 15 p 18 15 p 19 15 p 20 15 p 21 15 p 22 15 p 24 15 p 4 14 p 7 14 p 10 14 p 12 14
p 15 14 p 17 14 p 18 14 p 21 14 p 22 14 p 23 14 p 8 13 p 12 13 p 14 13 p 17 13 p 20 13 p 21 13 p 22 13 p 4 12 p 5 12 p 7 12 p 8 12 p 9 12 p 10 12 p 11 1
2 p 13 12 p 14 12 p 15 12 p 17 12 p 18 12 p 24 12 p 12 11 p 14 11 p 16 11 p 21 11 p 4 10 p 5 10 p 6 10 p 7 10 p 8 10 p 9 10 p 10 10 p 12 10 p 13 10 p 16
10 p 18 10 p 22 10 p 23 10 p 4 9 p 10 9 p 13 9 p 14 9 p 15 9 p 16 9 p 17 9 p 19 9 p 21 9 p 22 9 p 4 8 p 6 8 p 7 8 p 8 8 p 10 8 p 12 8 p 16 8 p 18 8 p 1
9 8 p 23 8 p 24 8 p 4 7 p 6 7 p 7 7 p 8 7 p 10 7 p 12 7 p 13 7 p 17 7 p 18 7 p 19 7 p 20 7 p 21 7 p 4 6 p 6 6 p 7 6 p 8 6 p 10 6 p 12 6 p 13 6 p 14 6 p
16 6 p 19 6 p 22 6 p 4 5 p 10 5 p 12 5 p 13 5 p 16 5 p 17 5 p 20 5 p 21 5 p 22 5 p 4 4 p 5 4 p 6 4 p 7 4 p 8 4 p 9 4 p 10 4 p 12 4 p 13 4 p 14 4 p 15 4
p 16 4 p 18 4 p 20 4 p 23 4 p
%%EOF

```

Output when qr is scanned with phone:

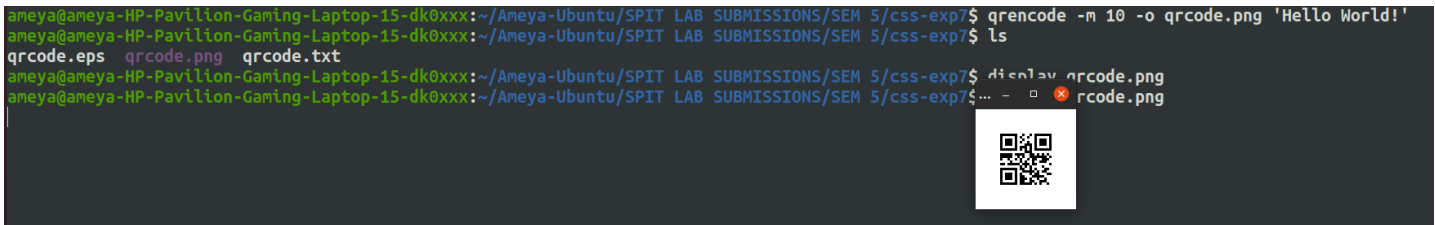




Outputting QR codes in ASCII characters just for fun. Each # corresponds to a dot.

In the QR code image, the software creates a white border the width of one dot. This facilitates the process of deciphering the code for programs or the smartphone later on. If desirable, you can increase or decrease the edge with the `-m` parameter; in the following example the border width would be 10 code pixels:

`$ qrencode -m 10 -o qrcode.png 'Hello World!'`



If you are saving the QR code in PNG format, the `-s` parameter specifies the height of a black QR code pixel. By default, Qrencode draws every black dot three by three pixels. The program creates a slightly smaller QR code with quite a wide margin with the following command:

```
$ qrencode -s 2 -m 10 -o qrcode.png 'Hello World!'
```



A white border appears around the QR code that is exactly 20 screen pixels wide (10 QR code dots in width, with each dot two pixels). If you are creating EPS images, the only parameter available is `-m`; `-s` moves the entire QR code out of the image in this case. To control the resolution of PNGs further, use the `-d` parameter to define the dpi(dots per inch).

DPI of 300



Tolerance

Besides the specified data, the QR code contains additional error correction information. If a portion of the image is damaged, it allows you to reconstruct the missing or illegible data. The more additional information the QR code contains, the more heavily damaged it can be without becoming useless.

Increasing the error tolerance increases the size of the image because you need more black dots



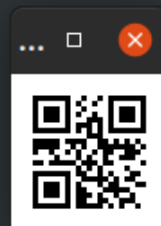
Both QR codes contain the text Hello World!; the one on the left uses the highest error correction level, H , and therefore is more robust – but also larger.

The QR code standard thus uses four levels of error correction: H level allows you to read all data if 30 percent of the QR code is destroyed, Q level if 25 percent is unreadable, and M level if just 15 percent is unintelligible. At the lowest level, L , only 7 percent of the data can be faulty. In Qrencode, the `-l` parameter selects the error correction level. The possible values are pretty much what you would expect: L , M , Q , and H .

Of the tested programs, Portable QR-Code Generator offers the greatest functionality. Data entry is also convenient in a variety of tabs. Qrencode creates QR codes more quickly, and it is easy to scripted so you generate QR codes in batches. KBarcode4-light is deprecated, but it is the only program that can generate PDF files; the EPS images created by Qrencode offer similar output.

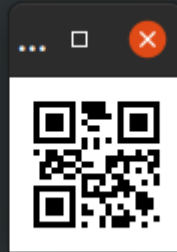
```
qrencode -l L -o qrcode.png 'Hello World!'
```

```
EM 5/css-exp7$ qrencode -l L -o qrcode.png 'Hello World!'  
EM 5/css-exp7$ display qrcode.png
```



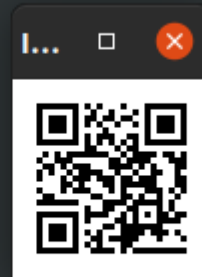
```
qrencode -l M -o qrcode.png 'Hello World!'
```

```
exp7$ qrencode -l M -o qrcode.png 'Hello World!'  
exp7$ display qrcode.png
```



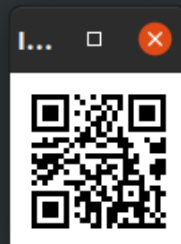
qrencode -l Q -o qrcode.png 'Hello World!'

```
s-exp7$ qrencode -l Q -o qrcode.png 'Hello World!'  
s-exp7$ display qrcode.png
```



qrencode -l H -o qrcode.png 'Hello World!'

```
5/css-exp7$ qrencode -l H -o qrcode.png 'Hello World!'  
5/css-exp7$ display qrcode.png
```



EXERCISE

TPO of SPIT has asked you to submit a simple QR code of your CV so that the company can make an offline process to create a short list of potential candidates. Hence as a part of exercise create CV showing your Profile, Ability and Capability and generate QR code of CV. This QR code can only be shared between you and the company. You will provide your digital signature as an authentication to the company. Once authentication is successful the company is able to scan this QR code of your CV.

Show Implementation steps that you will generate QR code and sign it with your digital signature. Share it with your colleague from the same lab who is capable of verifying your digital signature. Successful verification makes scanning of QR code feasible. Show you creativity in terms of robustness of generating digital signature, generating of QR code and Scanning of QR code if a relevant secret key is entered.

Creating new keys for encryption

```
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/
resume-qr$ gpg --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Ameya Jangam
Email address: ameya.jangam@spit.ac.in
You selected this USER-ID:
    "Ameya Jangam <ameya.jangam@spit.ac.in>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 2AC142DD36F649BB marked as ultimately trusted
gpg: revocation certificate stored as '/home/ameya/.gnupg/openpgp-revocs.d/9BD2058F3A9A3BCBF6D60D5B
2AC142DD36F649BB.rev'
public and secret key created and signed.

pub   rsa3072 2021-12-14 [SC] [expires: 2023-12-14]
       9BD2058F3A9A3BCBF6D60D5B2AC142DD36F649BB
uid           Ameya Jangam <ameya.jangam@spit.ac.in>
sub   rsa3072 2021-12-14 [E] [expires: 2023-12-14]
```

Exporting newly generated keys to public and private key

```
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/
resume-qr$ gpg --export -a "Ameya Jangam"> ameya_public.key
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/
resume-qr$ gpg --export-secret-key -a "Ameya Jangam"> ameya_private.key
```

Encoding the URL of the Resume using qrcode

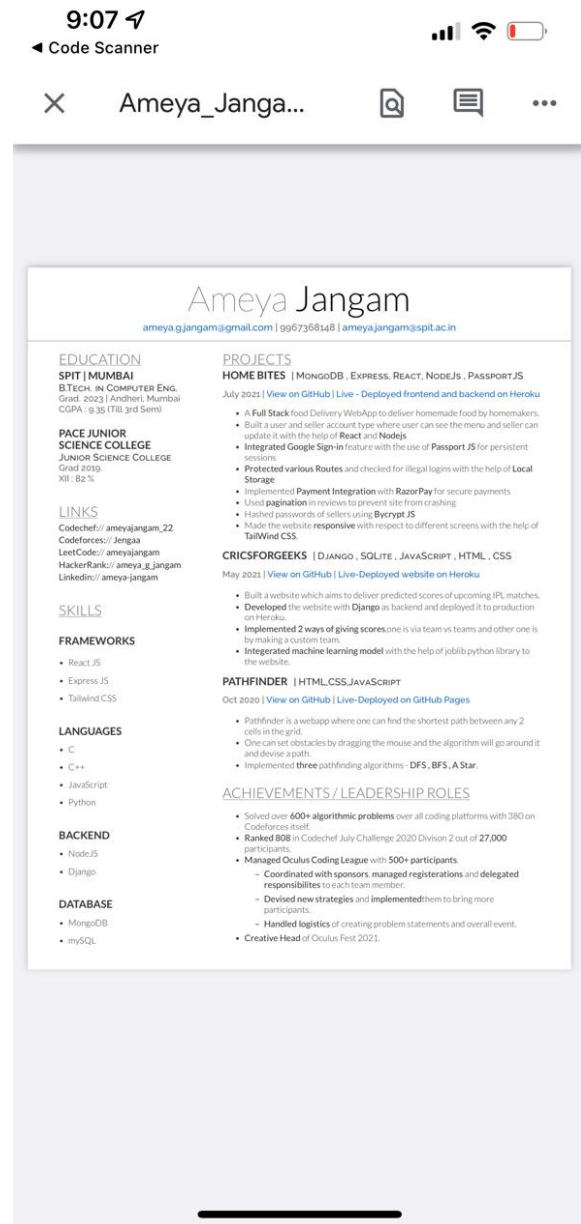
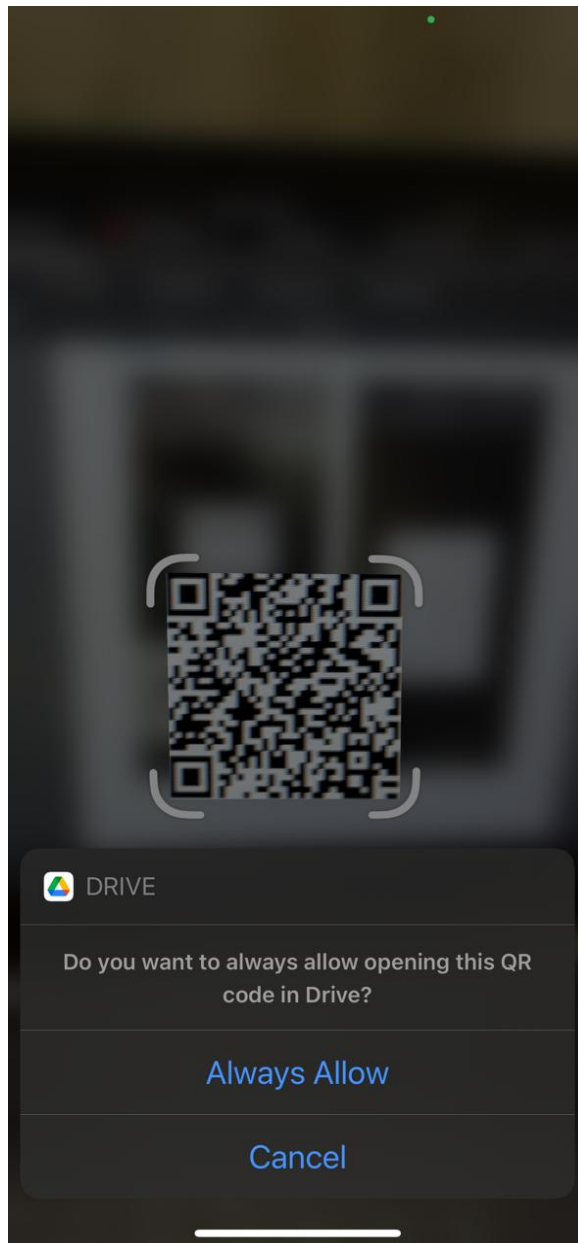
```
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/
resume-qr$ qrencode -o resume_ameya_jangam.jpg https://drive.google.com/file/d/1hDMdy0faL7RWYQNkuFn
MGH0n9cyilBsA/view?usp=sharing
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/
resume-qr$ gpg -e -a -u "Ameya Jangam" -r "Gajendra Jangam" resume_ameya_jangam.jpg
File 'resume_ameya_jangam.jpg.asc' exists. Overwrite? (y/N) y
```

Once encrypted using the keys of Ameya Jangam, user Gajendra is able to decrypt the same using his own keys. This makes sure that only the intended users are able to access the resume from the QR Code.

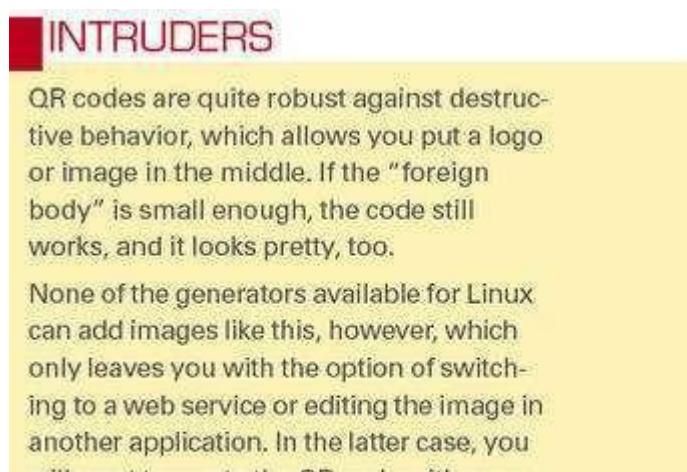
```
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/GPIT LAB SUBMISSIONS/SEM 5/css-exp7/  
resume-gr$ qrencode -o resume_ameya_jangam.jpg https://drive.google.com/file/d/1hDMdy0faL7RWYQNkuFn  
MGH0n9cyilBsA/view?usp=sharing  
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/  
resume-gr$ gpg --e -a -u "Ameya Jangam" -r "Gajendra Jangam" resume_ameya_jangam.jpg  
File 'resume_ameya_jangam.jpg.asc' exists. Overwrite? (y/N) y  
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/  
resume-gr$ gpg --decrypt resume_ameya_jangam.jpg.asc  
gpg: encrypted with 3072-bit RSA key, ID B4522B808F11B5D1, created 2021-12-14  
      "Gajendra Jangam <gajendrajangam@gmail.com>"  
  
❖PNG  
❖  
IHDR❖)❖IPLTE❖❖❖❖tRNS❖❖❖    pHYs  
  
          ❖~❖IDATH❖❖❖1❖❖ ❖(❖❖  
  
:d❖❖/'❖_❖k qQ❖ ❖❖❖❖H❖❖D❖bPd❖❖                q  
❖❖❖'Rf❖❖Jj❖❖PnSe^O❖❖❖❖H❖❖6❖6d❖4❖❖❖❖❖4q❖❖!mti&❖❖❖[A~❖h❖❖❖{+❖#lZ❖❖t❖Jqm❖❖G{De❖pm❖4T"❖3❖_❖^q❖  
R4❖de'l❖❖|etgu eysv❖D❖❖❖❖>ec#/❖❖i❖eb`8❖❖}er+❖#❖'eh3❖❖H❖So,❖er❖❖'.❖zSLW❖❖>Fe❖Y❖❖❖^U❖❖❖%rT❖Z  
❖S❖❖qu❖z8❖8  
        ❖D❖❖Fza2?3&❖V❖❖RA{!❖>❖l❖  
  
❖YoX❖❖P  
❖❖❖PkeP❖❖]Fg❖F❖f<I*"❖=❖❖❖❖❖❖❖❖❖❖❖❖❖/❖U'❖❖}❖Cpe❖U+❖❖w❖xm❖❖RIEND❖B`❖ameya@ameya-HP-Pavilion-  
Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/resume-gr$
```



Checking the output of scanning the QR code on an external device (Phone)



Following is for your information only, but if you would like to try you can give a go....



The level you choose depends on the proposed use. If you expect the QR code to be exposed to external influences (e.g., wind, weather, graffiti) or if you are planning to manipulate the image deliberately (see the “Intruders” box), you will want the highest level, *H*. On a business card, the lowest level, *L*, is sufficient. If in doubt, simply omit the *-l* parameter. The program then uses level *L* by default.

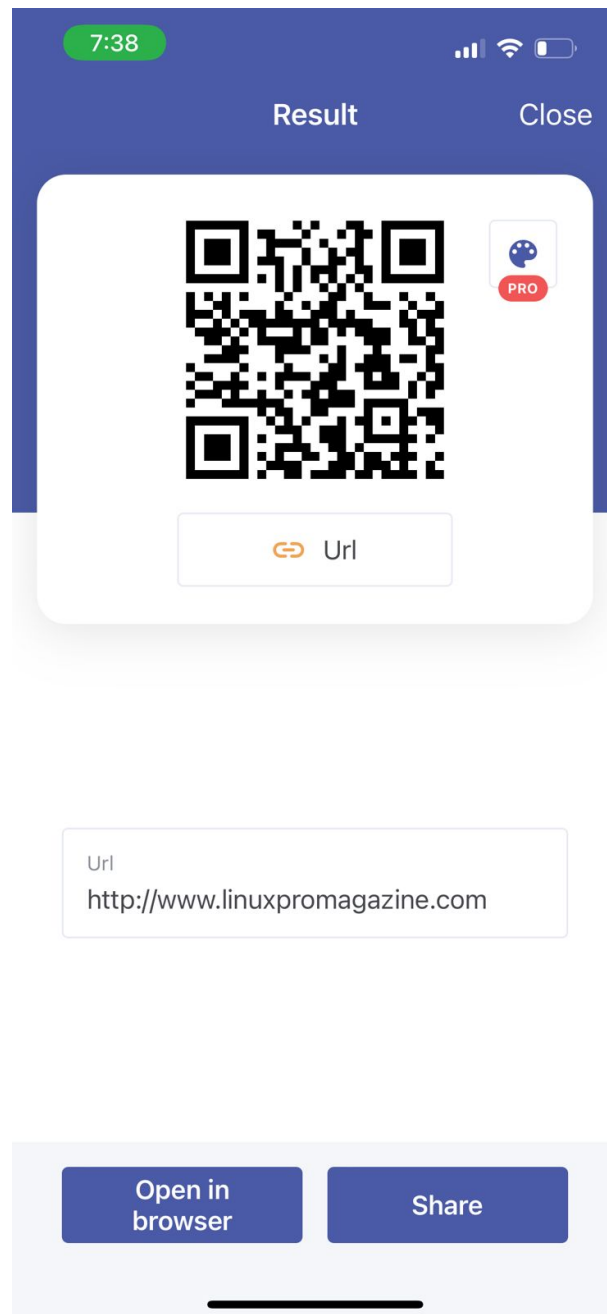
Variety

Qrcode lets you encode any text, including URLs. You only need to use single quotes if blanks and non-standard characters appear in the text:

```
$ qrcode -o lpm.png http://www.linuxpromagazine.com
```

All modern QR code readers recognize that this is a web address and will offer to open it in the browser.

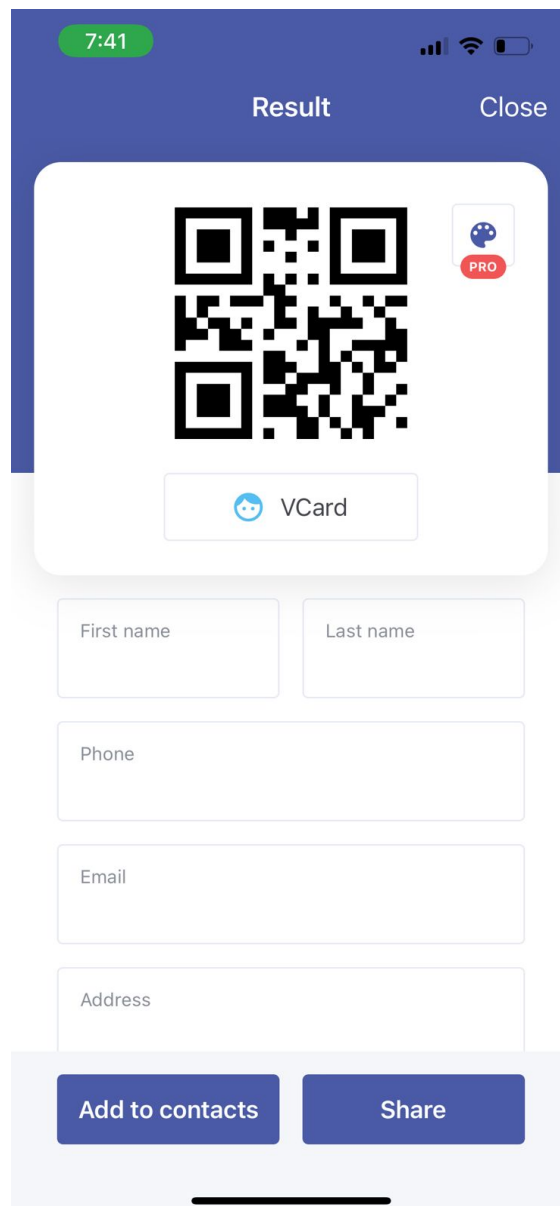




Getting an QR-encoded address automatically to enter a smartphone owner's address book is more difficult. In this case, you need to enter your address and some additional, cryptic information

```
$ qrencode -o address.png 'BEGIN:VCARD VERSION:4.0 FN: N:Schuermann;Tim;;;  
ADR:;;Putzbrunner Str. 71;Munich;;81739; END:VCARD'
```

```
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/  
qrcode-2$ qrencode -o address.png 'BEGIN:VCARD VERSION:4.0 FN: N:Schuermann;Tim;;; ADR:;;Putzbr  
unner Str. 71;Munich;;81739; END:VCARD'  
ameya@ameya-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/Ameya-Ubuntu/SPIT LAB SUBMISSIONS/SEM 5/css-exp7/  
qrcode-2$ display address.png
```

A terminal window with a dark background and green text. The command `qrencode -o address.png 'BEGIN:VCARD VERSION:4.0 FN: N:Schuermann;Tim;;; ADR:;;Putzbrunner Str. 71;Munich;;81739; END:VCARD'` has been executed. Below the terminal, a small window titled 'I...' displays a square QR code.

Typing this is not exactly easy, so to make life easier, you might prefer a QR code generator with a graphical

user interface, such as [QtQR](#).

Sign-In
to create Dynamic QR Codes

QR Code

Generate
Static QR Code

Scan
Using your webcam

About

What's a QR Code?

Privacy

Terms of use

Imprint

QR Code Generator

URLFREE TEXTCONTACTAPPPHONESMS

Name
Jangam

First Name
Gajendra

Organization
CIDCO

Email
gajendra7@gmail.com

Phone
9999999999

cell

Fax

Street
Near Grand Central Mall, Seawoods

City
Navi Mumbai

Region
-

Postcode
400706


Country
India

URL/Website
https://youtu.be/cT7Sgjllo0-s

MAKE DYNAMIC

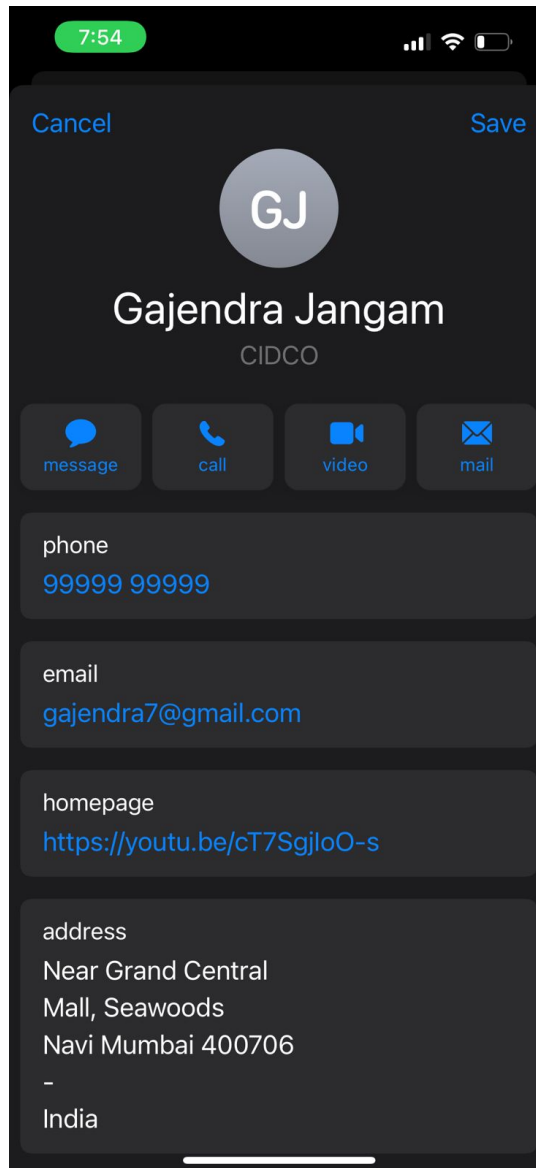
Static QR Code

Content is directly encoded in the image. If you want to be able to update it later, create a Dynamic QR Code.



Generated QRCode





Of the tested programs, Portable QR-Code Generator offers the greatest functionality. Data entry is also convenient in a variety of tabs. Qrcode creates QR codes more quickly, and it is easy to scripted so you generate QR codes in batches. KBarcode4-light is deprecated, but it is the only program that can generate PDF files; the EPS images created by Qrcode offer similar output.

Vulnerabilities: <https://analyticsindiamag.com/25-years-of-qr-codes-a-look-at-vulnerabilities-of-this-popular-payment-method/>

Currently, over 23% of Trojans and viruses are transmitted via QR codes. On the 25th anniversary of QR codes, its creator, Masahiro Hara wants to make QR scanning more secure.

Usually, in the case of QR scanning, possible scenarios of attacks can be summarised as follows:

- QR codes cannot be hacked. One way hackers to infiltrate this system by changing the QR code added in the poster. These fake posters can be circulated in the public domains and clueless customers scan these fake QR codes and end up visiting phishing websites.

- This usually happens because of the increase in the number of mobile users. Mobiles make it hard to verify the full link in the address bar. This makes users more vulnerable. When they use this phishing page to login, their passwords are compromised.
- An attacker might set up a fake website and redirect users by changing the QR Code. This is dangerous if some form of credentials are needed to access the website. The user has no possibility to verify that the link is not modified.
- SQL injection is another form of attack that occurs when SQL queries are made with user input text inserted into the query string. QR code readers are subject to data injection into their structured objects when they attempt to interpret the data of a QR code.
- A malicious party can create a QR code that injects arbitrary strings into a user's data structures potentially causing harm to the user.
- Criminals can simply prepare malicious QR codes and affix them over legitimate codes which may result in victims inadvertently making payments to a criminal rather a legitimate service provider.
- QRLJacking or Quick Response Code Login Jacking is a simple social engineering attack vector capable of session hijacking affecting all applications that rely on "Login with QR code" feature as a secure way to login to accounts.
- [QRLJacking attack](#) gives attackers the ability to apply a full account hijacking scenario on the vulnerable Login with QR Code feature resulting in accounts stealing and reputation affection.

Conclusion:

1. QR Codes are a quick way to transfer information from one user to another.
2. QR codes can be encrypted and decrypted, so only authorized users can access the information.
3. QR codes have error correction and tolerance levels and information can be extracted from code even if part of the code is corrupted.

Github Link: <https://github.com/ameyajangam22/CSS-LAB-2019130025>