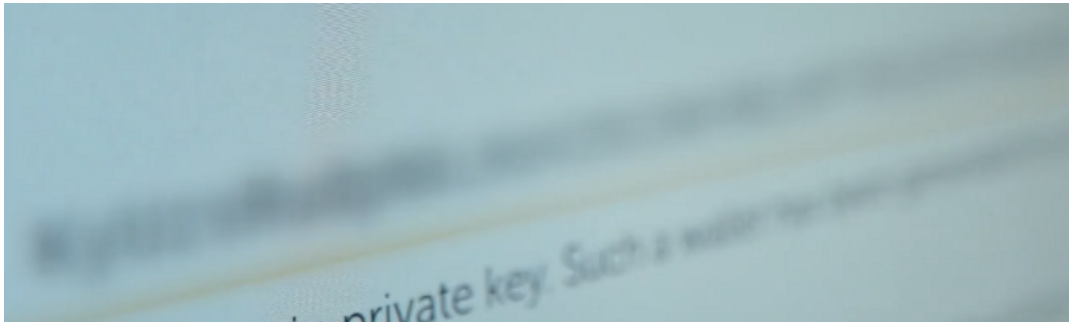
@rogerkver's $1,000 wallet obfuscated private key

« Malheursement nous ne sommes pas un jeu télévisé... »

https://youtu.be/hN0_1pU4U5w?t=1040

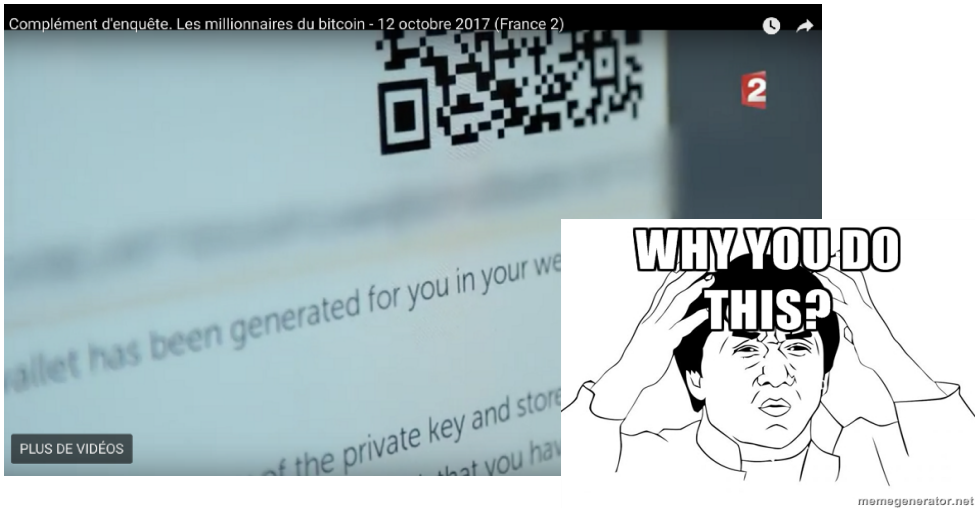
JOUONS…

Collecte d’informations



Clé privé : « kyU?sR...»

Clé publique



Une partie du qr code[priv] en clair

Clé privé + qr code[priv] : «...V»

Le QRcode...

© 2012 Walter Tuvell

QR Code — Structure

Model 2005 — ISO/IEC 18004:2006

Cell

- 0_2 ↔ "Light"
- 1_2 ↔ "Dark"

} **Bits ↔ Modules** (nominal color/reflectance)

Version ($1 \le V \le 40$): Size=$N \times N$; $N = 4 \cdot V + 17 \Rightarrow N^2$ modules

Quiet Zone: $4N_M$; ≥4-mod light margin surrounding cell

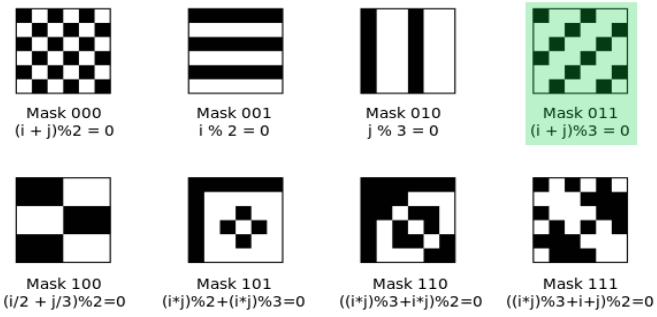
Function Patterns

- **Separators:** $6N_M$; $1 \times 8 \Rightarrow 45$ mods
- **ID/Finder/Positioning/Orientation:** $3N_M$; $7 \times 7 \Rightarrow 147$ mods
- **Alignment:** $K\,N_M$; $K \in \{0,1,6,13,22,33,46\}$; $5 \times 5 \Rightarrow 25 \cdot K$ mods
- **Timing:** $2N_M$; $1 \times (N-16) \Rightarrow 2 \cdot (N-16)$ mods

Encoding Region

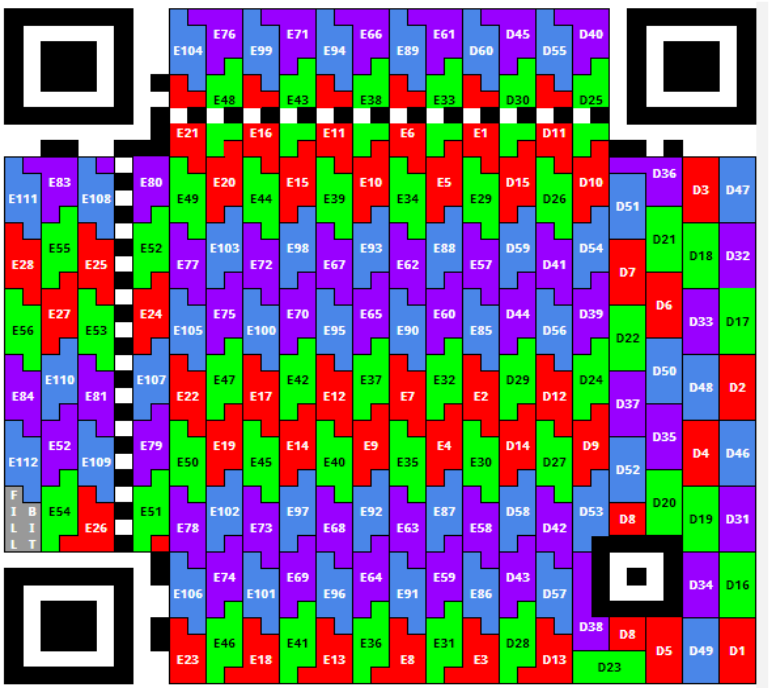
- **Format Info:** $5N_M$; 1×8, 1×6, $3 \Rightarrow 31$ mods
- **Version Info:** $2N_M$; $V \ge 7$ only; $3 \times 6 \Rightarrow 0 \veebar 36$ mods
- **Content:** Cell \ Artifacts = Data(&Pads) + EDC(&Remainder)
  E.g.: $V=3 \Rightarrow 29^2 - (45+147+25+26+31+0) = 567$ content mods

Version	Error correction level	Number of data codewords[a]	Number of data bits[b]	Data capacity			
				Numeric	Alphanumeric	8-bit Byte	Kanji
5	L	108	864	255	154	106	65
	M	86	688	202	122	84	52
	Q	62	496	144	87	60	37
	H	46	368	106	64	44	27
6	L	136	1 088	322	195	134	82
	M	108	864	255	154	106	65
	Q	76	608	178	108	74	45
	H	60	480	139	84	58	36



...est robuste aux erreurs

- ECC : Reed-Solomon
  - Ajout de redondance d'information
  - Robuste aux erreurs en rafale
  - couplé à un entrelaceur et codeur convolutif



15 mots de donnés

28 mots de codes correcteurs

On liste les bits connus

4 bits	8 bits	416 bits	4 bits	48 bits
Mode Indicator	Character count indicator	Message (Private Key)	Terminator (4 bits)	Padding bits
0100	00110100	XXXXXXXXXXXXXXXXXXXXXXXX	0000	11101100 00010001 11101100 00010001 11101100 00010001

		Binary	Décimal			Binary	Décimal
Data Block 1	Codewords #1	0100 0011	67	Data Block 3	Codewords #31	0111 0011	115
	Codewords #2	0100 0100	68		Codewords #32	xxxx 0xxx	?
	Codewords #3	1011 0111	183		Codewords #33	xxxx 0xxx	?
	Codewords #4	0010 0101	149		Codewords #34	0000 0110	6
	Codewords #5	0101 0111	87		Codewords #35	xxxx 0xxx	?
	Codewords #6	1010 0111	167		Codewords #36	xxxx 0xxx	?
	Codewords #7	0011 0101	53		Codewords #37	xxxx 0xxx	?
	Codewords #8	0010 0111	39		Codewords #38	1110 0101	229
	Codewords #9	xxxx 0xxx	?		Codewords #39	xxxx 0xxx	?
	Codewords #10	xxxx 0xxx	?		Codewords #40	xxxx 0101	?
	Codewords #11	xxxx 0xxx	?		Codewords #41	xxxx 0xxx	?
	Codewords #12	xxxx 0xxx	?		Codewords #42	0010 0111	39
	Codewords #13	1011 0100	180		Codewords #43	0110 0101	101
	Codewords #14	xxxx 0xxx	?		Codewords #44	xxxx 0xxx	?
	Codewords #15	xxxx 0xxx	?		Codewords #45	xxxx 0111	?
Data Block 2	Codewords #16	0101 0101	85	Data Block 4	Codewords #46	xxxx 0xxx	?
	Codewords #17	xxxx 0xxx	?		Codewords #47	xxxx 0xxx	?
	Codewords #18	xxxx 0xxx	?		Codewords #48	xxxx 0xxx	?
	Codewords #19	xx00 0011	?		Codewords #49	0010 0110	38
	Codewords #20	0001 0xxx	?		Codewords #50	xxxx 0xxx	?
	Codewords #21	xxxx 0xxx	?		Codewords #51	xxxx 0xxx	?
	Codewords #22	xxxx 0xxx	?		Codewords #52	xxxx 0xxx	?
	Codewords #23	0110 0011	99		Codewords #53	0011 0101	53
	Codewords #24	xxxx 0xxx	?		Codewords #54	0110 0000	96
	Codewords #25	xxxx 0xxx	?		Codewords #55	1110 1100	236
	Codewords #26	xxxx 0xxx	?		Codewords #56	0001 0001	17
	Codewords #27	xxxx 0xx0	?		Codewords #57	1110 1100	236
	Codewords #28	0110 0011	99		Codewords #58	0001 0001	17
	Codewords #29	xxxx 0xxx	?		Codewords #59	1110 1100	236
	Codewords #30	xxxx 0xxx	?		Codewords #60	0001 0001	17

Bits deduced from image analysis and ASCII table Protocol related bits Bits read on the QR code

Limite de correction : E + 2*T ≤ D – P
E : nombre ecrasement (erasures)
T : nombre d'erreurs (errors)
D : nombre de mots de corrections
P : 0 dans ce cas

Capacité de correction 28 octects par Blocks

Block 1 : 6 erasures data, 22 erasures ECC
Block 2 : 12 erasures data, 21 erasures ECC
Block 3 : 10 erasures data, 18 erasures ECC
Block 4 : 6 erasures data, 21 erasures ECC

=> On peut tout recupérer sauf Block 2

Recupérations des blocks

```python
import reedsolo

#Initialize the codec for a number of 28 EC codewords
rs = reedsolo.RSCodec(28)

# Message/Data codewords array, value of each codeword in decimal
mess = [115, 0, 0, 6, 0, 0, 0, 229, 0, 0, 0, 39, 101, 0, 0]
# Error correction codewords array, value of each codeword in decimal
ecc = [0, 58, 199, 0, 0, 0, 40, 147, 0, 0, 0, 15, 254, 0, 0, 0, 65, 138, 0, 0, 0, 166, 0, 0, 0, 0, 0, 38]
# Error position array, contains index of the erasures
error_pos = [1, 2, 4, 5, 6, 8, 9, 10, 13, 14, 15, 18, 19, 20, 23, 24, 25, 28, 29, 30, 33, 34, 35, 37, 38, 39, 40, 41]
# Decode the message and indicate the decoder the position of the errors
w = rs.decode(mess+ecc, error_pos)
# Print the decimal value of the recovered message/data codewords
print [x for x in w]

# Result
[115, 22, 181, 6, 151, 103, 118, 229, 22, 133, 167, 39, 101, 164, 87]
#
```

Le block 3 contient 33 erreurs → on va bruteforcer 21 bits qui permettent de construire les 5 octects qui manquent.
ça nous fait 2 097 152 de possibilités (ça prend ~ 30 mins)

→ donne 2 résultats possibles.

Et c’est gagné

KyUzsRudpNkLKeV2815KV9EzRf7EG1kPivwnQhZrvZEwhKrbF7CV

Bonus :
https://blockchair.com/bitcoin-cash/outputs?q=recipient(17Qgadvc7pm51mV9r9zUAs4xU1XXwDRr8o)

Sources :
https://medium.freecodecamp.org/lets-enhance-how-we-found-rogerkver-s-1000-wallet-obfuscated-private-key-8514e74a5433