

EXPLORE GO CRYPTOGRAPHY

"I laughed, I learned. It's a great book."

—Dian Amencourt

b d f i j k m n q s u v

w z b d f i j k m n q s

u v E X P L O R

d f i j k G O

C R Y P T O G R

u v w z b d f i j k m n q s
v w x y z i j

Go 1.22



A GOPHER'S GUIDE TO CIPHERS

JOHN ARUNDEL

Contents

Praise for <i>Explore Go: Cryptography</i>	12
Introduction	13
Cryptography	13
Why learn about cryptography?	14
What do you need to know?	14
About the book	15
What you'll learn	15
How to use this book	16
1. Ciphers	17
Codes and ciphers	17
A shift cipher	18
Key points	19
Wheels within wheels	19
Enciphering	19
One byte at a time	20
Assume a spherical cow	20
From behaviour to test	21
Choose your own adventure	22
A first test	22
Anatomy of a test	23
When should the test fail?	23
The world's greatest bug detector	23
A function about nothing	24
Cutting code	25
The smarty-pants answer	26
We're gonna need a bigger test	27
A table of cases	27
Combining cases	27
A suspiciously similar test	28
A test case struct	28
Looping over the cases	29
Lighting a fire under the test	30
Introducing subtests	30
Subtests and <code>t.Run</code>	30
A litany of failure	32
Adding more cases	32
A working table test for <code>Encipher</code>	33

2. Enciphering	36
A simple enciphering filter	36
A command package	37
Garbage in, garbage out	37
All about Eve	38
Reverse engineering	38
Variable keys	39
Adding a keyhole	39
New keys, new cases	40
Compiler complaints	43
Stochastic debugging	43
Getting back to green	44
Getting the key	46
Defining a key flag	46
Getting the flag value	46
Enciphering with arbitrary keys	47
3. Deciphering	49
Brute force and ignorance	49
A statistical vulnerability	50
How many possible keys are there?	50
How safe is your safe?	51
Doing it for the crack	51
Cribs	52
Designing a cracking function	52
First, we need Decipher	52
Do we even need a test, then?	54
New test, same old table	55
Look before you loop	56
A deciphering tool	57
Let's not complicate encipher	57
4. Cracking	60
Testing a Crack function	60
Failure is an option	61
A key-guessing function	62
A command-line cracker	63
What do users want?	63
Crib constraints	64
Cracking a real ciphertext	65
Waiting for the tiger	65
The devil in the details	66
0x89 is the magic number	67
Printing the unprintable	67
5. Keys	69

Lost in keyspace	69
The space of all possible keys	70
How long should the key be?	70
A keyspace the size of the universe	71
Multi-byte keys	72
Embiggening the key	72
Updating the test cases	72
Bytes in, bytes out	73
Fixing Encipher	73
A magical solution	73
Constraining the key index	74
Modulate your enthusiasm	74
The modulus operator	75
Refactoring will continue until morale improves	76
Testing longer keys	76
Designing the test cases	77
Sharpening the tools	78
When the right answer is wrong	78
What kind of flag do we need?	79
The joy of hex(adecimal)	80
Binary notation	80
A string flag	81
Where's the BEEF?	81
Changing frequencies	82
6. Cribs	84
Cracking long keys	84
What needs to change?	85
A byte at the problem	85
Length limitations	86
Checking our guesses	86
Knowing when to stop	87
A proper little cracker	87
Updating the crack tool	89
Applying brute force	89
Crib considerations	90
A false deciphering	90
On being the right size	90
7. Passwords	92
Security	92
Password spaces	93
Dictionary words	93
Extremely guessable passwords	94
Passphrases	94
Rules	95

Character sets	95
Password policies	95
You're really not helping	96
Maximising the keyspace	97
Unicode	97
Generation	98
What does "random" mean?	98
Predictability	99
Binary choices	99
Information	100
Knowledge	100
8. Blocks	102
Block ciphers	102
Why blocks?	103
The cipher.Block interface	104
Plugs and sockets	104
Adapting the shift cipher	105
The Encrypt method	105
Where's the key?	105
Fixing the key size	105
A cipher constructor	106
The NewCipher function	107
If the key fits	107
A polite refusal	108
A special error value	108
Wrapping errors	109
Writing NewCipher	109
Implementing the cipher.Block interface	110
Testing Encrypt	110
Circumstances alter cases	111
Creating the cipher object	111
A test for Encrypt	112
Writing Encrypt and Decrypt	113
Forgotten something?	114
9. Modes	116
Handling data in blocks	116
Updating the encipher tool	117
Block by block	117
The cipher.BlockMode interface	118
A simple block mode	118
Testing CryptBlocks	119
Creating the encrypter object	120
Implementing CryptBlocks	121
Using a BlockMode in encipher	123

Block alignment	124
When misaligned data attacks	124
Checking our assumptions	124
The long and short of it	125
A test about nothing	125
A panic-proof CryptBlocks	126
10. Padding	128
A padding scheme	128
Making ends meet	129
The “illegal pixel” problem	129
An unambiguous padding scheme	129
Testing Pad	130
Implementing Pad	132
Writing Unpad	133
Padding input to encipher	134
Adding the padding	135
Checking the results	135
Deciphering	136
Adding the unpadding	136
A decrypter that implements BlockMode	137
Waiting for the tiger	139
11. Enumeration	141
Adventures in keyspace	141
A new test for cracking long keys	141
A block is a black box	142
Enumerating long keys	144
Designing a Next function	144
Testing Next	145
Big endians and little endians	146
A simple implementation	146
Checking our key guesses	147
The soul of a new cracker	148
We apologise for the delay	148
Benchmarking key cracking	149
Pick an easier problem	149
Understanding benchmark output	150
Cranking up the difficulty	150
Ballparking a realistic crack time	151
A test we might live to see pass	151
Updating the crack tool	152
Putting it all together	152
Will it crack?	153
Parallelisation	154
Big computers are too slow	154

Dense computers become black holes	155
Parallel cracking is still useful	155
But we won't implement it here	155
Strong keys can't be cracked	156
12. Entropy	157
Information	158
Entropy: a measure of our ignorance	158
Entropy of digital messages	159
Compression	159
Enciphered data	160
Identifying ciphertexts	160
Complexity	160
Describing sequences	161
A generating algorithm	161
The shortest program that describes a sequence	162
Kolmogorov complexity	162
Complexity from simplicity	163
Security	164
Randomness is high entropy	165
Key generation	165
Pragmatic non-determinism	165
A little entropy goes a long way	166
Being unpredictable	166
13. Randomness	168
Pseudo-random generation	168
Randomness in games	169
The Fibonacci sequence	169
A simple random generator	169
A Fibonacci generator in Go	170
Repeatability	171
Seeding the RNG	171
Security problems	172
Periodicity	172
Distribution	172
Uniformity	173
"Secure enough" random generators	173
Environmental noise	173
The system entropy pool	173
Security is relative	174
You don't need to outrun the bear	174
Keeping secrets from the gods	175
Hardware entropy sources	175
Defeating "God Emperor Eve"	175
Quantum measurements	176

A quantum randomness source	176
Generating quantum keys	177
14. Chains	178
Visualising the problem	178
A completely random image	179
Separating metadata and pixel data	181
Hidden figures	181
Muddying the waters	183
What's wrong with this picture?	184
Mallory in the middle	185
ECB: a block-headed operating mode	185
Introducing Mallory	185
Block replay	186
Dropping and modifying blocks	186
More sophisticated modes	186
Counter mode (CTR)	187
Nonces	187
Cipher Block Chain (CBC)	187
Initialization Vector (IV)	188
Generating a secure IV	188
Implementing CBC mode	189
Enciphering in CBC mode	189
The updated encipher program	190
Updating the decipher program	192
The devil in disguise	193
15. Hashing	195
Message integrity	195
The night is dark, and full of errors	196
Bit-flipping and block-dropping	196
Integrity checking	197
Digests and hashing	197
Hash tables	197
Buckets and distribution	198
Collisions	198
Cryptographic hashing	199
Simple hash functions	199
A naïve algorithm	199
Implementing LenHash	200
Problems with LenHash	201
Preimage attacks	202
A slight improvement	202
Implementing SumHash	202
Testing SumHash	203
A preimage attack on SumHash	204

The avalanche effect	204
Real hash algorithms	205
MD5	205
SHA-1	205
SHA-256	206
Password hashing	206
Zero-knowledge secret storage	206
Password hashing requirements	207
Rainbow tables	207
Salting	208
Slow down, you move too fast	208
16. Coins	209
Cryptocurrency	209
Let there be “Bobcoin”	210
The problem of trust	210
A distributed digital ledger	211
Security	211
The double-spending problem	211
Ordering transactions	212
The blockchain	212
Doomed stubs	213
Integrity	213
Consensus	214
Proof of work	214
Up in smoke	215
Proof of stake	215
17. Authentication	216
Message integrity	216
Hash preimage attacks	217
Chosen ciphertext attacks	217
MAC	218
Length extension attacks	218
HMAC	219
Key exchange	219
The problem	219
Key splitting	220
Asymmetric encryption	220
Public and private keys	221
The Diffie-Hellman-Merkle protocol	221
A one-way function for key exchange	222
A DHM worked example	223
The reveal	223
Public-key cryptography	224
RSA	225

Signing	225
Authentication	226
Verification	226
The chain of trust	227
18. Cryptography	228
A little history	229
The origins of DES	229
Tripling down on DES	229
AES	230
Rijndael	230
The starting grid	230
Round and round	231
Confusion and diffusion	231
Putting it all together	232
Implementing AES	233
Getting ready to rumble	233
Ding ding, round one	234
The diffusion loop	234
The last round	235
AES encryption in practice	237
Enciphering with AES-CBC	237
Updating encipher and decipher	237
Encryption plus authentication	241
Enciphering with AES-GCM	241
The final final version	242
Weaknesses	245
Implementation fails	246
Mashing the keys	246
Tomorrow	247
The future is quantum	247
Quantum parallelism	247
The catch	248
Why your computer isn't quantum... yet	249
Post-quantum cryptography	249
Afterword	250
About this book	251
Who wrote this?	251
Feedback	251
Free updates to future editions	252
Join my Go Club	252
For the Love of Go	252
The Power of Go: Tools	253
Further reading	253
Credits	253

Praise for *Explore Go: Cryptography*

I laughed, I learned. It's a great book that manages to explain complicated ideas in really simple, straightforward language.

—Dian Amencourt

One of the most accessible and even fun books about cryptography I've ever read. John writes vividly, with many a neat turn of phrase, and the text is full of wit and humour.

—Kurtis Connor

This is a remarkable book, practically fizzing with enthusiasm for its subject. It whisks you through the worlds of cryptography, mathematics, physics, and computer science, and shows you that they're all connected in surprising and thought-provoking ways. Highly enjoyable.

—Cuong Quoc

As a software engineer who works on a security product (I won't say which one), I've always felt like a bit of an imposter when it comes to cryptography: I hoped no one would ever ask me how a cipher actually works. After reading this book, I now feel a lot more confident. Thanks, John!

—Yuliana M

So friendly and easy to read. This has really unlocked a few things for me. I can't recommend this book highly enough!

—Joseph Adewale

John has a knack of writing almost luminously clear explanations that stick in your mind long after finishing the book.

—Patrick Devlin