Exposed Cipher

h	q1	0
t	d2	1
)	r1	2
-	f0	3
d	r5	4
E	m0	5
b	t7	6
S	j2	7
u	c1	8
P	p3	9
"	m8	@
I	c9	.
J	r7	_
L	h7	-
A	z5	:
8	n1	#
3	s7	/
N	r3	
{	s0	(
(	g0	)
C	n5	[
m	d6	]
#	c8	|
1	x2	?
G	v3	!
K	x5	;
7	f3	"
O	q8	*
9	q2	<
}	j4	>
'	i1	,
U	t3	%
`	r9	$
n	f8	=
*	i7	a
^	c6	b
6	i4	c
g	f6	d
M	w9	e
+	l5	f
]	l7	g
Q	y6	h
!	g5	i
X	p8	j
T	g6	k
o	m3	l
F	o0	m
Z	z6	n
p	e4	o
V	x4	p
:	a1	q
i	k6	r
a	t2	s
>	x3	t
<	q7	u
l	e3	v
|	d9	w
%	p4	x
W	m1	y
,	j9	z
f	j8	A
.	k1	B
2	j0	C
=	l8	D
@	o5	E
D	p9	F
$	w4	G
/	k0	H
r	f9	I
~	w0	J
s	a0	K
&	x9	L
k	j5	M
_	n7	N
[	r0	O
H	k2	P
?	g3	Q
R	i9	R
c	c2	S
q	t5	T
4	b1	U
e	p1	V
;	t9	W
j	d5	X
\	l4	Y
5	v0	Z
Y	v5	'
0	m7	}
B	l3	{

Exposed Cipher Decoder Test Instructions

Goal:	Your objective is to review/modify the exposed data in this sheet, including the cipher, the offset key, the masked output and the decoder formula to determine if there is a vulnerability in the exposed data that can be used to decode the statement in **G18**.
Locate the Cipher Data:	Columns A–C contain an Exposed Cipher. This is the mapping logic used in the formula to determine the encoding you see in in the masked output in **F19**
Find the Offset Key:	Cell **F18** contains the Offset Key used to shift / manipulate the cipher. You may use this value as needed to test different decoding attempts
Check the Masked Output:	Cell **F19** contains the Masked Output (the encoded text). This is what you will attempt to decode into readable text. Note: This would be embedded in the decoder formula as the fullStr but is included here as an external reference for this exercise
Review the Decoder Formula:	Cell **E23** displays the **Expo**sed Decoder Formula. This same formula is actively applied in **G18** to generate a decoding attempt.
Password	Cell **F20** is the password field that, when entered correctly, will decode the text in **G18.** The password is not stored in this workbook and is intentionally withheld for this exercise
Test the Decoder:	Observe the result in **G18**. If the decoder is correct, **G18** will display legible text/phrase without any extraneous characters or symbols.

Exposed Data	Encoder Inputs	Decoder Output
Offset Key	**asdkjflakdf jlkdsdfjlke sdsldkflke lksdfjeksleikemjsdl**	**TNMHFtMBHGcKNGcMAxcPHKDyEHPcLHcBwxtLcvtG cuxcINMcMHcPHKDj**
Masked Output*	**k[e\tYGm^!C2_[aS-u\!!><%&;P/mK)^MZiR:Rj8;('N%sRVdS_U**	
Enter Password		

Full Decoder Formula in G18

```
TEXTJOIN("",TRUE,
LET(
stateset3,IF($F$18<>"",$F$18,"jJrsSjhFR:B.H>?1HoiFqY$Kbri"),
fullStr,IF($F$19<>"",$F$19,"u.;gA:SMYsUfAg.3kGFLJ"),
offsetStr,IF($F$18<>"",$F$18,"H4\7JGFk3.f(88LQoH_,si+"),
password,IF($F$20<>"",$F$20,"dTQ#}'&gS{s##e>=-1R"),

offsetNum,SUMPRODUCT(CODE(MID(offsetStr,ROW(INDIRECT("1:"&LEN(offsetStr))),1))),
baseOffset,MOD(offsetNum,89)+1,

maskMult,MOD(
SUMPRODUCT(CODE(MID(offsetStr,ROW(INDIRECT("1:"&LEN(offsetStr))),1)))+LEN(offsetStr),
MOD(LEN(offsetStr),SUMPRODUCT(CODE(MID(stateset3,ROW(INDIRECT("1:"&LEN(stateset3))),1))*LEN(stateset3)))+SUMPRODUCT(CODE(MID(stateset3,ROW(INDIRE
CT("1:"&LEN(stateset3))),1))*LEN(stateset3))
)+5,

maskOffset1,MOD(offsetNum+LEN(offsetStr),89)+1,
maskOffset2,MOD(offsetNum*maskMult,89)+1,
maskOffset3,MOD((offsetNum+maskMult)*LEN(offsetStr),89)+1,
maskOffset4,MOD(offsetNum*(maskMult+LEN(offsetStr)),89)+1,

passwordMask,MOD(SUMPRODUCT(CODE(MID(password,ROW(INDIRECT("1:"&LEN(password))),1)))*LEN(password),89),
totalMask,MOD(maskOffset1+maskOffset2+maskOffset3+maskOffset4+passwordMask,89),

restored,IF(AND($F$18="",$F$19="",$F$20=""),
fullStr,
LEFT(fullStr,LEN(fullStr)-2)&RIGHT(fullStr,1)
),

n,LEN(restored),
seq,SEQUENCE(n),

charMask,IF(ISODD(seq),
MOD(maskOffset1+maskOffset3+passwordMask,89),
MOD(maskOffset2+maskOffset4+passwordMask,89)
),

chars,MID(restored,seq,1),

compRows,MAP(chars,LAMBDA(c,
XLOOKUP(TRUE,EXACT(c,$A$2:$A$90),ROW($A$2:$A$90)-ROW($A$2)+1)
)),
cipherRows,MOD(compRows-baseOffset-totalMask-MOD(seq*3,89)-seq+89*4,89)+1,
cipherTokens,INDEX($B$2:$B$90,cipherRows),

cipherTokenRows,MAP(cipherTokens,LAMBDA(t,
XLOOKUP(TRUE,EXACT(t,$B$2:$B$90),ROW($B$2:$B$90)-ROW($B$2)+1)
)),
charRows,
MOD(
cipherTokenRows
-baseOffset
-totalMask
-MOD(seq*3,89)
-seq
+IF(AND($F$18="",$F$19="",$F$20=""),seq*7,89*4),
89)+1,

INDEX($C$2:$C$90,charRows)
)
)
```