Base64 Encryption & Minecraft Geoguessr

Mystiz

Base64 Encryption (Crypto, 6 solves)

Challenge Summary.

> *People said that base64 is an encoding, not an encryption. Did they have a misconception about that?*
>
> *If you believe that base64 is just an encoding, then convince me that you are able to "decode" the article (which is in English).*

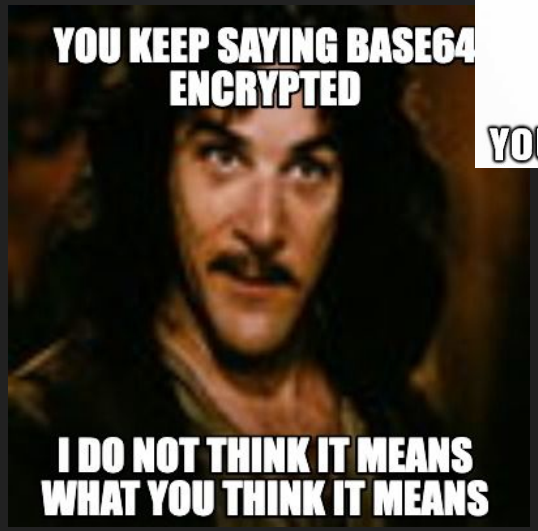

```
HiNWLMEhnoJQRDjKyz4ln3AW0OE6nzx6RD4hYwjqLidhna65YbQQRo4CYwZqRMVqYwj5Yw4CVihT+ihO
Ywd6VoJ6nO1lYw65YoJP0Hc5VoxeyHcC0UcT+wrW0OE6nzx6RD4hYwSDYwd6VoJ6nO1WRMYW0MECVzc5
YwSDYwd6VoJ6nO1W+iNWLHcG+zcP0zET0zQTFUcr+wrWRixT+wSeYw65Yox50iXWLz1WLiNWLi6eYoJC
YwE30ijB+ihOYw4KLz45+i4QRbcG+zcP0zE5FWPpJOE6nzx6RD4hYwjqLidhna65Yw65YwEQnaxeYwSq
YoJP0HcDLi4TYoJPLzXKYw6qYwjqyHcO+z06RUc5VoE6Vw4PYwSDYoV3+zJT0iNWRwjq0MxQ0arKYw46
nOJQ+iNWRwxTVwx3n3cQRDXWLaSZLD6qLzJlRah5YwSDYwd6VoJ6nO1WRa4GVzYWVa6T+bcaLzEh+ihO
Yw030zj20ihG+ix5FUc4RME6RM06nUAWVwQ6nDrW+z1WLHcG+wj3Li4T0zElnMJlL3ce+z4TnD6UVzJl
RaNWRaLWRwxTVwx3n3cT+wjTYw65YoECViVPRoeWVwQ6Yo4QRirW0DS3YwjKRiS5VbcQRwAWnajZnwd6
n3cC0UcT+wjTYwdQRDV2LiV6FUcwRMYW+ih5VwjqLarKYwVlVDxqYwfWnaxGVw6CRUcC0UcjRDVK+z4P
YwdQRDV2LiV6FbcjFbcrFbccYwjq0bcvYwj30HcT+wrWRiS5VbcGRa2ZRaNKYoVP+id6YjPKYjfKYjWW
LiheYfPWLzE6YoEQnDrqYfdl+axM+z46FbcrHbAWJxYKYfStFbcQRDXWXrNWLzE6YoJP0HcZRM4TYw4C
Ri2CRUcALi63n3cC0UcK0zJT0zE5YbQT0zEZ0iXWLD6OnDjZn3cCnUce+iV3LzcPn3eKYwjq0bcsr3AW
JrrKYiJrFbcQRDXWJeLWLzE6YoJP0HcZRM4TYw4CRi2CRUc30zc6LzJ5FUcr+wrWRDSqnaxqnarWnwO3
```

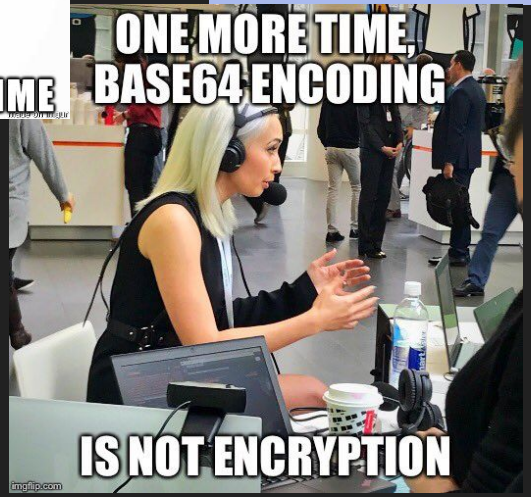
Base64 Encryption (Crypto, 6 solves)

Solved by…

- *2 tertiary teams,*
  - *T0024: HKUST* 🥇
  - *T0055: HKUST*
- *3 open teams, and*
  - *O0031: AUTOEXEC.BAT*
  - *O0056: T0003 (2022) @ TWY's Temple*
  - *O0045: select * from flags*
- *1 invited team*
  - *Project Sekai*

Base64 Encryption (Crypto, 6 solves)





Base64 Encryption (Crypto, 6 solves)

The character set is

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdef
ghijklmnopqrstuvwxyz0123456789+/
```

…but it is shuffled!

In short, this is a substitution cipher, but the message is first encoded in base64.

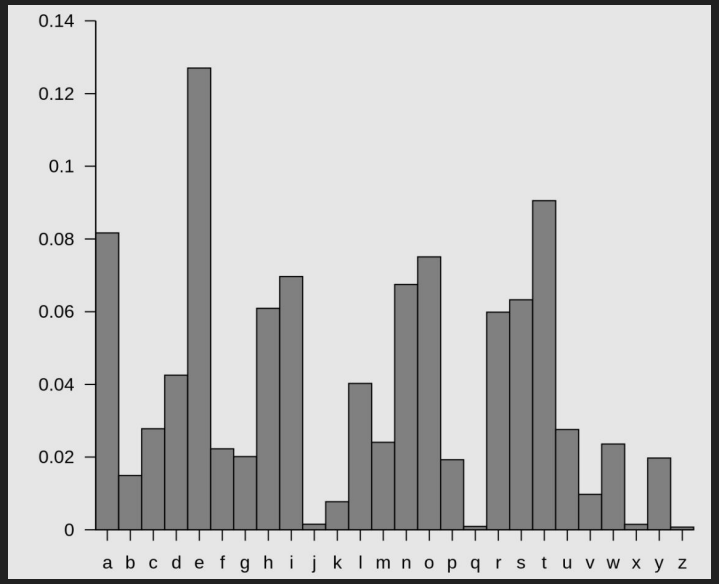
Base64 Encryption (Crypto, 6 solves)

In substitution ciphers, each letter is converted into another letter.

💡 ***Observation.*** If A (message) is encrypted to b (ciphertext), the numbers of A's and the numbers of b's are the same. This happens to every character pairs.

The frequency is unchanged! We can look at the frequency of the letters to what encrypts to what.

For example, if w appears the most in the ciphertext, w probably is encrypted from E.



Base64 Encryption (Crypto, 6 solves)

Base64: Three characters are encoded into four characters in base64.

Source	Text (ASCII)	M							a								n							
	Octets	77 (0x4d)							97 (0x61)								110 (0x6e)							
Bits		0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Base64 encoded	Sextets	19							22					5				46							
	Character	T							W					F				u							
	Octets	84 (0x54)							87 (0x57)					70 (0x46)				117 (0x75)							

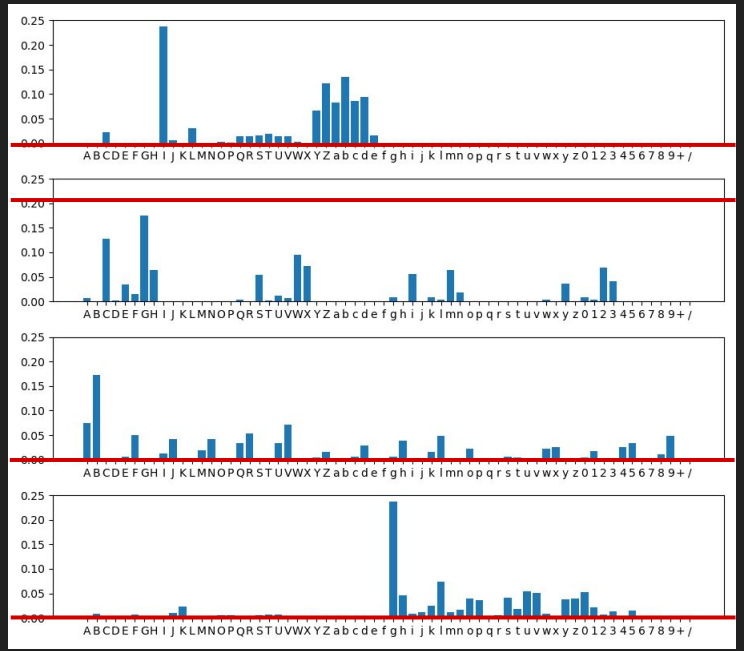
Can we apply frequency analysis on this one?

Base64 Encryption (Crypto, 6 solves)

Ground truth: *Shakespeare's text*

Assuming that our ciphertext has a similar distribution to Shakespeare's text (not as elegant tho):

- The character occurred the most is w, occurring at 0%, 20.5%, 0% and 0.3% in the four groups respectively.

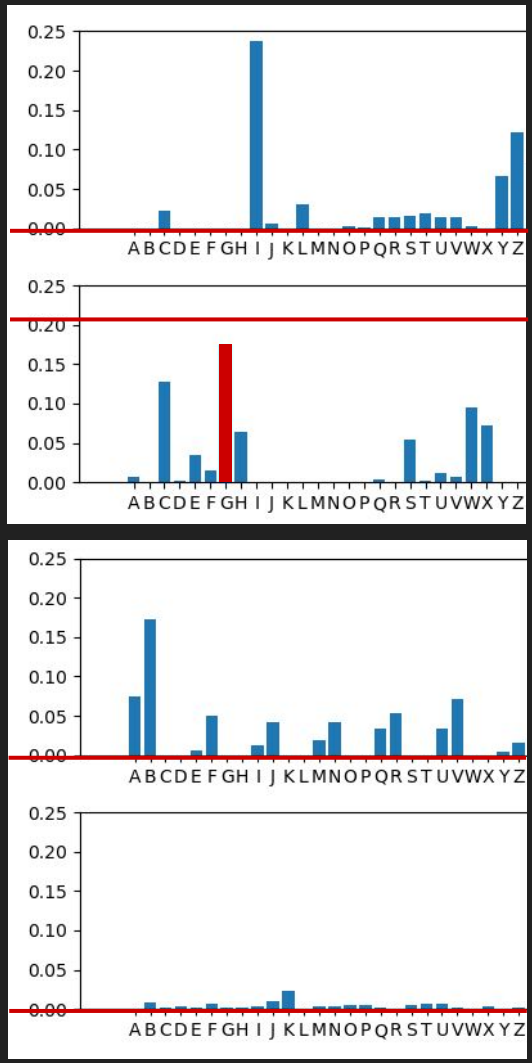


Base64 Encryption (Crypto, 6 solves)

Ground truth: *Shakespeare's text*

Assuming that our ciphertext has a similar distribution to Shakespeare's text (not as elegant tho):

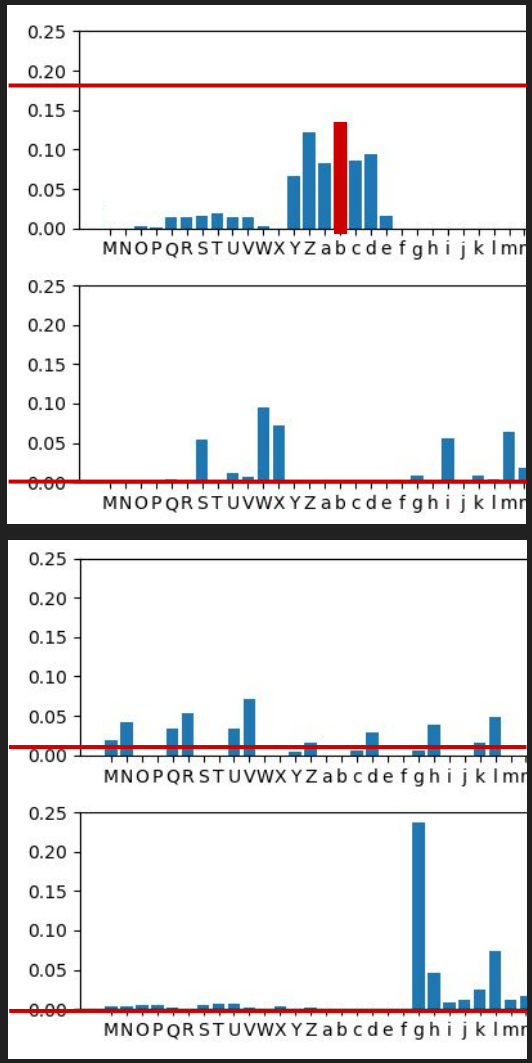
- The character occurred the most is w, occurring at 0%, 20.5%, 0% and 0.3% in the four groups respectively.
- The frequencies are similar to G, with frequencies being 0%, 17.6%, 0%, 0.2% respectively.
- Let's assume w is G encrypted.



Base64 Encryption (Crypto, 6 solves)

- The second character occurred the most is Y, occurring at 18.5%, 0%, 1.0% and 0% in the four groups respectively.
- The frequencies are similar to b, with frequencies being 13.6%, 0%, 0%, 0.03% respectively.
- Let's assume Y is b encrypted.

…Now you get the idea. Repeating the process 64 times, we have a mapping between the *ciphertext* and the *plaintext*. (Can be done with programs!)



Base64 Encryption (Crypto, 6 solves)

This is what we have…

```
.lertyiuhfalx .k,%wtah/af`9latqcyhxom*(ep3lh.fv4laol`3ann-nwlcaauakt.lhomuit@oa.ux@sv
n,bewtah/af`9ln&lcaauakt%fv.vts` oln&lcaauakt%-lep@m, it$nt*.N I,beeon,n5lhom.ouc%p$%q
leqhumusldruak-nwl`,p /-`(e m, it$oNZzWtah/af`9latqcyhxolholdhh.uln4muit@fq`.muip#,lht
latx@w,'af oata``)ln&mv2,%nuledatv.htrll`!kuh-ledon`ork hfc%p~#rhtp%Jd~oln&lcaauakt%dp
-`&.`xn- Gp$y-nwlgrt!dunm-ooN @ftafwaj,%`j!jbe,$%p@m,arq`.t$JjuJs u, .jhb`%Jd|edw%don`
ork n,anlhomtsaf)e}e`j!mp(ebevn2lalen/a hdl%hqchcak sv n,anlchff$qf!N Ffv.-no`atprllf.
botl`.h.m`hsf sv Eff,, )lchff$qf!M EM IM Alatu {lart@n,been/a mdy#d|lmv)-calZllP.lYeqn
ul:ep$amthjbtl3J,.g, !M I.,%W..l>?M hfc%0.ep$amuit@cfp.l`3ei3f pqhrk sv lt%nt$om*.t$cu
c%rhwjack sj u-f2p ik-llatu N',%T.llUIM hfc%WW%p$amuit@cfp.l`3ei3f rt ap%oN I,befn4h.t
hrehj2p !m$E.1{..e&..T3}n rt rt !guomuitG.B cfp.lgrt!dunnlcaauakt%-leaxp-`(e Eff,, )lc
hff$qf!muays4JR.f ody!l`.hj!kt,mp$pyeitshor`hak sv n,befan`$he lqnwaawt@pdaJguays%p$am
prt !kwau Jf n,bepxp,or`oqa,%qnumuit !mphauajnolj(bbe`j!mps`ot`hhe ng bt@aypldxnuc%-le
p@m, it$nt*.Mn4e}ep%nq`+N Qv yfrep$alabdbe`j.pqoa n,bep%nq`+l`3ktarulxL%yn$mv.dl%rbeqd
lt@ng wjabmuit@fdaw. i,p!ks2Cx..t/.t.{.9.w0`$O@ ..nO@~m@c.f.Of....t..trtyk..@nxN
```

Base64 Encryption (Crypto, 6 solves)

More improvements on the heuristics?

Article is in English ⇒ MSB for each byte is unset.

- characters in group 1 cannot be `ghijklmnopqrstuvwxyz0123456789+/`,
- characters in group 2 cannot be `IJKLMNOPYZabcdefopqrstuv456789+/`,
- characters in group 3 cannot be `CDGHKLOPSTWXabefijmnqruvyz2367+/`.

Source	Text (ASCII)	M							a								n							
	Octets	77 (0x4d)							97 (0x61)								110 (0x6e)							
Bits		0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Base64 encoded	Sextets	19							22								5							46
	Character	T							W								F							u
	Octets	84 (0x54)							87 (0x57)								70 (0x46)							117 (0x75)

Base64 Encryption (Crypto, 6 solves)

More improvements on the heuristics?

Article is in English ⇒ MSB for each byte is unset.

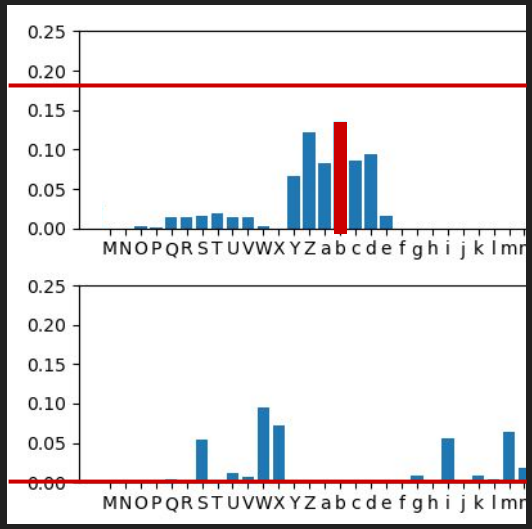
- characters in group 1 cannot be `ghijklmnopqrstuvwxyz0123456789+/`,
- characters in group 2 cannot be `IJKLMNOPYZabcdefopqrstuv456789+/`,
- characters in group 3 cannot be `CDGHKLOPSTWXabefijmnqruvyz2367+/`.

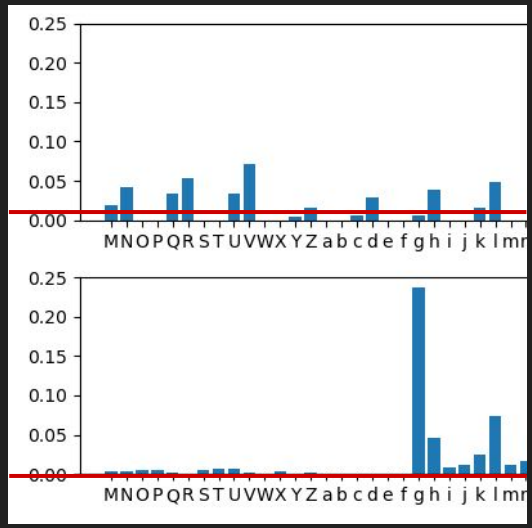
We can add a **penalty** to improve the heuristic.

Base64 Encryption (Crypto, 6 solves)

REWIND

- The second character occurred the most is Y, occurring at 18.5%, 0%, 1.0% and 0% in the four groups respectively.
- The frequencies are similar to b, with frequencies being 13.6%, 0%, 0%, 0.03% respectively.
- Wait… Y cannot map to b because it appeared in group 3, which is not allowed in English.

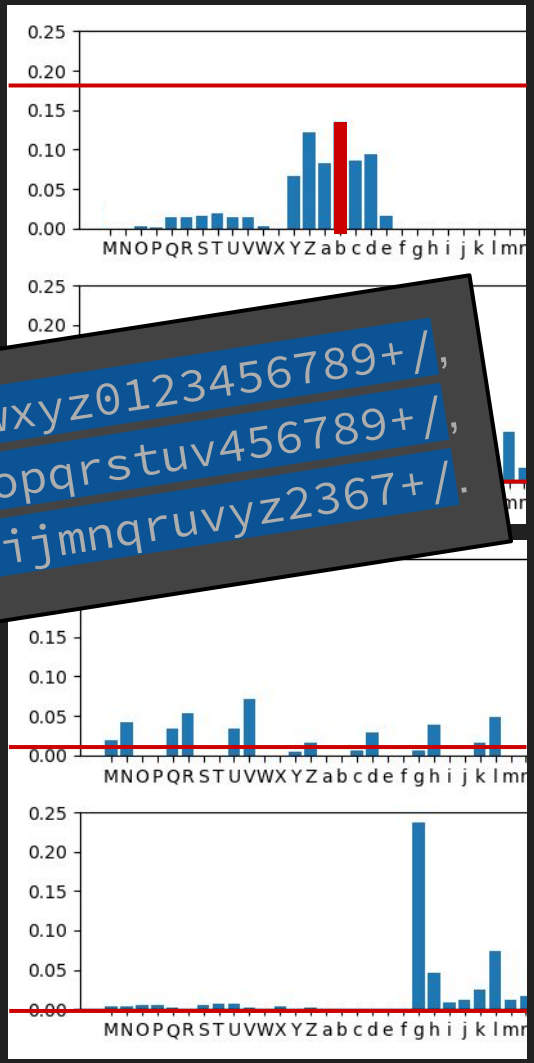




Base64 Encryption (Crypto, 6 solves)

REWIND

- The second character occurred the most is
  occurring at 18.5%, 0%
  four
  characters in group 1 cannot be ghijklmnopqrstuvwxyz0123456789+/,
  characters in group 2 cannot be IJKLMNOPYZabcdefopqrstuv456789+/,
  characters in group 3 cannot be CDGHKLOPSTWXabefijmnqruvyz2367+/.
  p to b because it appeared in
  group 3, which is not allowed in English.



Base64 Encryption (Crypto, 6 solves)

This is what we have…

```
.lertyiuhfalx .k,%wtah/af`9latqcyhxom*(ep3lh.fv4laol`3ann-nwlcaauakt.lhomuit@oa.ux@sv
n,bewtah/af`9ln&lcaauakt%fv.vts` oln&lcaauakt%-lep@m, it$nt*.N I,beeon,n5lhom.ouc%p$%q
leqhumusldruak-nwl`,p /-`(e m, it$oNZzWtah/af`9latqcyhxolholdhh.uln4muit@fq`.muip#,lht
latx@w,'af oata``)ln&mv2,%nuledatv.htrll`!kuh-ledon`ork hfc%p~#rhtp%Jd~oln&lcaauakt%dp
-`&.`xn- Gp$y-nwlgrt!dunm-ooN @ftafwaj,%`j!jbe,$%p@m,arq`.t$JjuJs u, .jhb`%Jd|edw%don`
ork n,anlhomtsaf)e}e`j!mp(ebevn2lalen/a hdl%hqchcak sv n,anlchff$qf!N Ffv.-no`atprllf.
botl`.h.m`hsf sv Eff,, )lchff$qf!M EM IM Alatu {lart@n,been/a mdy#d|lmv)-calZllP.lYeqn
ul:ep$amthjbtl3J,.g, !M I.,%W..l>?M hfc%0.ep$amuit@cfp.l`3ei3f pqhrk sv lt%nt$om*.t$cu
c%rhwjack sj u-f2p ik-llatu N',%T.llUIM hfc%WW%p$amuit@cfp.l`3ei3f rt ap%oN I,befn4h.t
hrehj2p !m$E.1{..e&..T3}n rt rt !guomuitG.B cfp.lgrt!dunnlcaauakt%-leaxp-`(e Eff,, )lc
hff$qf!muays4JR.f ody!l`.hj!kt,mp$pyeitshor`hak sv n,befan`$he lqnwaawt@pdaJguays%p$am
prt !kwau Jf n,bepxp,or`oqa,%qnumuit !mphauajnolj(bbe`j!mps`ot`hhe ng bt@aypldxnuc%-le
p@m, it$nt*.Mn4e}ep%nq`+N Qv yfrep$alabdbe`j.pqoa n,bep%nq`+l`3ktarulxL%yn$mv.dl%rbeqd
lt@ng wjabmuit@fdaw. i,p!ks2Cx..t/.t.{.9.w0`$O@ ..nO@~m@c.f.Of....t..trtyk..@nxN
```

Before

Base64 Encryption (Crypto, 6 solves)

This is what we have…

```
.n%rtyuuhnaox -w,%gtat$!n`9 anqbytxs!*(mp, i.nv. as `,antinw baauaws- hs!uid@sat5x@lf
the%gtat$!n`9 cf baauaws%nr%ftl` s cf baauaws%in%p@ch id$td*4. The%md4hcu hs!t3elep#%q
n%qhu!ul dreakinw `/p 3i`(m ch id$s.Z.Wtat$!n`9 anqbytxs hs dhtt5 cn!uid@fq`4!uip,o hn
 anx@wh&an sata``) cf!v2h%ten%lanft(du/ `!wuhin%ld4`d2w hnlepsdrhnp%ml~s cf baauaws%lp
#`"%`xti Gp$yinw frd!|encid3. @ntanvav,%`j!ve%h#%p@charq`4d$mvums uh 4vhb`%ml~%lw%ld4`
d2w that hs!tlaf)m}e`j!!p(me%fcr aomcsa hll%tqdtbaw lf that bhnf<qf!. Fnr%ins`anpu/ f-
bd. h%tt#`hln lf Enf/h ) bhnf<qf!- E- T- A ane { ard@the%mcsa cl.$l~/!v)iba Zo X/ Yeqn
u".ep$a!thve.".mht'h !- T.,%U./".N- hnle.N%p$a!uid@dnp4 `,mo,n pqhrw lf od%td$s!*4d$de
lerhwvadw lv uif2p iw-o ane zS,%UE/ UT- hnleWW%p$a!uid@dnp4 `,mo,n rd ap%s. The%ncntt.
tu%tj2p !!$EB.{.N%SJ.V.P" rd rd !ous!uidOI2 dnp4 frd!|ent baauaws%in%axpi`(m Enf/h ) b
hnf<qf!!uay|nJ[Qn sl.! `-tj!ws/!p<pyeutltd2`haw lf the%nat`$hm oqnwaawd@plamouay|ep$a!
prd !wvae mn the%pxphd2`d8a,%qnu!uid !!phauavns j(be%`j!!pl`d.`hhm to bd@aypolxtelein%
p@ch id$td*4-cnm}ep%tq`+. Qf ynu%p$a able%`g.pqsa the%p%tq`+ `,wtaruoxL%yc|!v-ll%re%qd
od@to wvab!uid@flaw: ihp!w|r3y".s3.s.x69.w0`$O0 ..nO0~c0lIn8On?g.5n.t.rtyt7I0nM.
```

After

Base64 Encryption (Crypto, 6 solves)

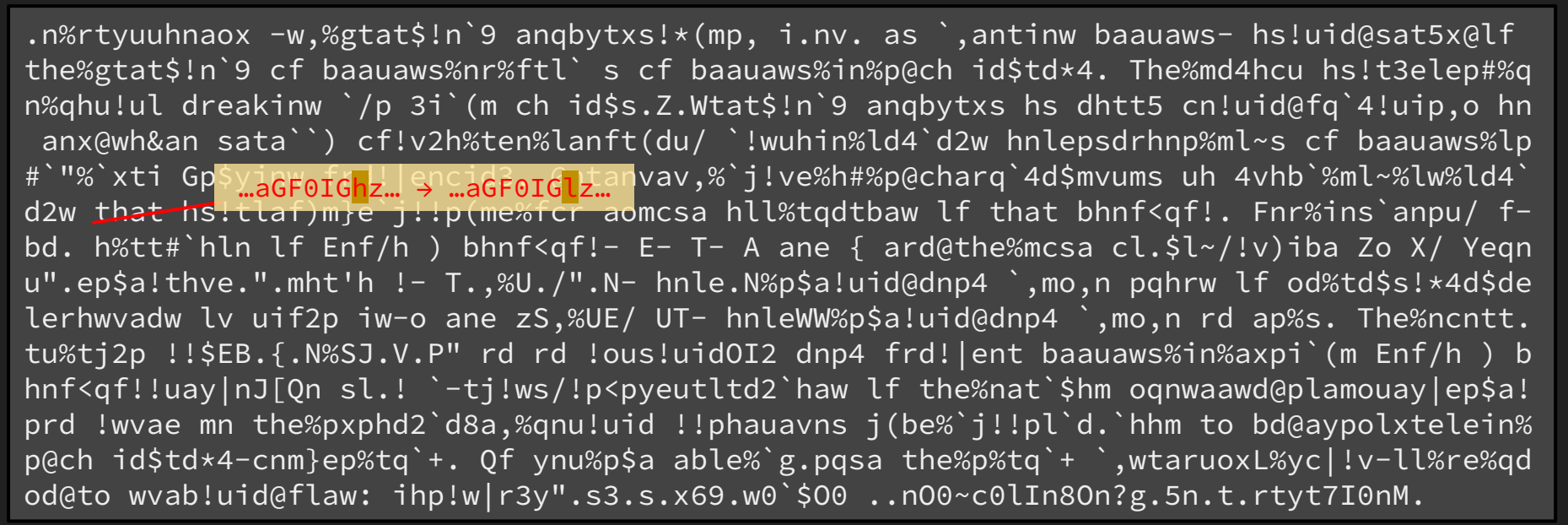
Patching word by word…

```
.n%rtyuuhnaox -w,%gtat$!n`9 anqbytxs!*(mp, i.nv. as `,antinw baauaws- hs!uid@sat5x@lf
the%gtat$!n`9 cf baauaws%nr%ftl` s cf baauaws%in%p@ch id$td*4. The%md4hcu hs!t3elep#%q
n%qhu!ul dreakinw `/p 3i`(m ch id$s.Z.Wtat$!n`9 anqbytxs hs dhtt5 cn!uid@fq`4!uip,o hn
 anx@wh&an sata``) cf!v2h%ten%lanft(du/ `!wuhin%ld4`d2w hnlepsdrhnp%ml~s cf baauaws%lp
#`"%`xti Gp$ jinw frd!|encid3. @ntanvav,%`j!ve%h#%p@charq`4d$mvums uh 4vhb`%ml~%lw%ld4`
d2w that hs!tlaf)m}e`j!!p(me%fcr aomcsa hll%tqdtbaw lf that bhnf<qf!. Fnr%ins`anpu/ f-
bd. h%tt#`hln lf Enf/h ) bhnf<qf!- E- T- A ane { ard@the%mcsa cl.$l~/!v)iba Zo X/ Yeqn
u".ep$a!thve.".mht'h !- T.,%U./".N- hnle.N%p$a!uid@dnp4 `,mo,n pqhrw lf od%td$s!*4d$de
lerhwvadw lv uif2p iw-o ane zS,%UE/ UT- hnleWW%p$a!uid@dnp4 `,mo,n rd ap%s. The%ncntt.
tu%tj2p !!$EB.{.N%SJ.V.P" rd rd !ous!uidOI2 dnp4 frd!|ent baauaws%in%axpi`(m Enf/h ) b
hnf<qf!!uay|nJ[Qn sl.! `-tj!ws/!p<pyeutltd2`haw lf the%nat`$hm oqnwaawd@plamouay|ep$a!
prd !wvae mn the%pxphd2`d8a,%qnu!uid !!phauavns j(be%`j!!pl`d.`hhm to bd@aypolxtelein%
p@ch id$td*4-cnm}ep%tq`+. Qf ynu%p$a able%`g.pqsa the%p%tq`+ `,wtaruoxL%yc|!v-ll%re%qd
od@to wvab!uid@flaw: ihp!w|r3y".s3.s.x69.w0`$O0 ..nO0~c0lIn8On?g.5n.t.rtyt7I0nM.
```

that is

Base64 Encryption (Crypto, 6 solves)

Patching word by word…



Base64 Encryption (Crypto, 6 solves)

Patching word by word…

```
.n%rtyuuhnaox -w,%gtet$%n`9 anqbytys!*(mp, i.nv. as `,antinw beauews- is!uid@sat5x@lf
the%gtet$%n`9 cf beauews%nr%ftl` s cf beauews%in%p@ch id$td*4. The%md4hcu is!t3elep#%q
n%qiu!ul dreakinw `/p 3i`(m ch id$s.Z.Wtet$%n`9 anqbytys is dhtt5 cn!uid@fq`4!uip,o in
 anx@wh&en sate``) cf!v2h%ten%lanft(du/ `%wuhin%ld4`d2w hnlepsdrinp%ml~s cf beauews%lp
#`"%`yti Gp$yinw frd!|encid3. @ntenvev,%`j%ve%h#%p@charq`4d$mvums uh 4vib`%ml~%lw%ld4`
d2w that is!tlaf)m}e`j%!p(me%fcr aomcsa hll%tqdtbew lf that bhnf<qf%. Fnr%ins`anpu/ f-
bd. h%tt#`iln lf Enf/h ) bhnf<qf%- E- T- A ane { ard@the%mcsa cl.$l~/!v)ibe Zo X/ Yeqn
u".ep$e!thve.".mht'h %- T.,%U./".N- hnle.N%p$e!uid@dnp4 `,mo,n pqirw lf od%td$s!*4d$de
leriwvadw lv uif2p iw-o ane zS,%UE/ UT- hnleWW%p$e!uid@dnp4 `,mo,n rd ep%s. The%ncntt.
tu%tj2p %!$EB.{.N%SJ.V.P" rd rd %ous!uidOI2 dnp4 frd!|ent beauews%in%aypi`(m Enf/h ) b
hnf<qf%!uey|nJ[Qn sl.% `-tj%ws/!p<pyeutltd2`iew lf the%nat`$hm oqnwaawd@plamouey|ep$e!
prd %wvee mn the%pyphd2`d8a,%qnu!uid %!phauevns j(be%`j%!pl`d.`ihm to bd@eypolytelein%
p@ch id$td*4-cnm}ep%tq`+. Qf ynu%p$e able%`g.pqsa the%p%tq`+ `,wteruoxL%yc|!v-ll%re%qd
od@to wvab!uid@flaw: ihp%w|r3y".s3.s.x69.w0`$O0 ..nO0~c0lIn9On?g.5n.t.rtyt7I0nM.
```

Base64 Encryption (Crypto, 6 solves)

Patching word by word…

```
.n%r                    x -w,%gtet$%n`9 anqbytys!*(mp, i.nv. as `,antinw beauews- is!uid@sat5x@lf
the%gtet$%n`9 cf beauews%nr%ftl` s cf beauews%in%p@ch id$td*4. The%md4hcu is!t3elep#%q
n%qiu!ul dreakinw `/p 3i`(m ch id$s.Z.Wtet$%n`9 anqbytys is dhtt5 cn!uid@fq`4!uip,o in
 anx@wh&en sate``) cf!v2h%ten%lanft(du/ `%wuhin%ld4`d2w hnlepsdrinp%ml~s cf beauews%lp
#`"%`yti Gp$yinw frd!|encid3. @ntenvev,%`j%ve%h#%p@charq`4d$mvums uh 4vib`%ml~%lw%ld4`
d2w that is!tlaf)m}e`j%!p(me%fcr aomcsa hll%tqdtbew lf that bhnf<qf%. Fnr%ins`anpu/ f-
bd. h%tt#`iln lf Enf/h ) bhnf<qf%- E- T- A ane { ard@the%mcsa cl.$l~/!v)ibe Zo X/ Yeqn
u".ep$e!thve.".mht'h %- T.,%U./".N- hnle.N%p$e!uid@dnp4 `,mo,n pqirw lf od%td$s!*4d$de
leriwvadw lv uif2p iw-o ane zS,%UE/ UT- hnleWW%p$e!uid@dnp4 `,mo,n rd ep%s. The%ncntt.
tu%tj2p %!$EB.{.N%SJ.V.P" rd rd %ous!uidOI2 dnp4 frd!|ent beauews%in%aypi`(m Enf/h ) b
hnf<qf%!uey|nJ[Qn sl.% `-tj%ws/!p<pyeutltd2`iew lf the%nat`$hm oqnwaawd@plamouey|ep$e!
prd %wvee mn the%pyphd2`d8a,%qnu!uid %!phauevns j(be%`j%!pl`d.`ihm to bd@eypolytelein%
p@ch id$td*4-cnm}ep%tq`+. Qf ynu%p$e able%`g.pqsa the%p%tq`+ `,wteruoxL%yc|!v-ll%re%qd
od@to wvab!uid@flaw: ihp%w|r3y".s3.s.x69.w0`$O0 ..nO0~c0lIn9On?g.5n.t.rtyt7I0nM.
```

the[space]

Base64 Encryption (Crypto, 6 solves)

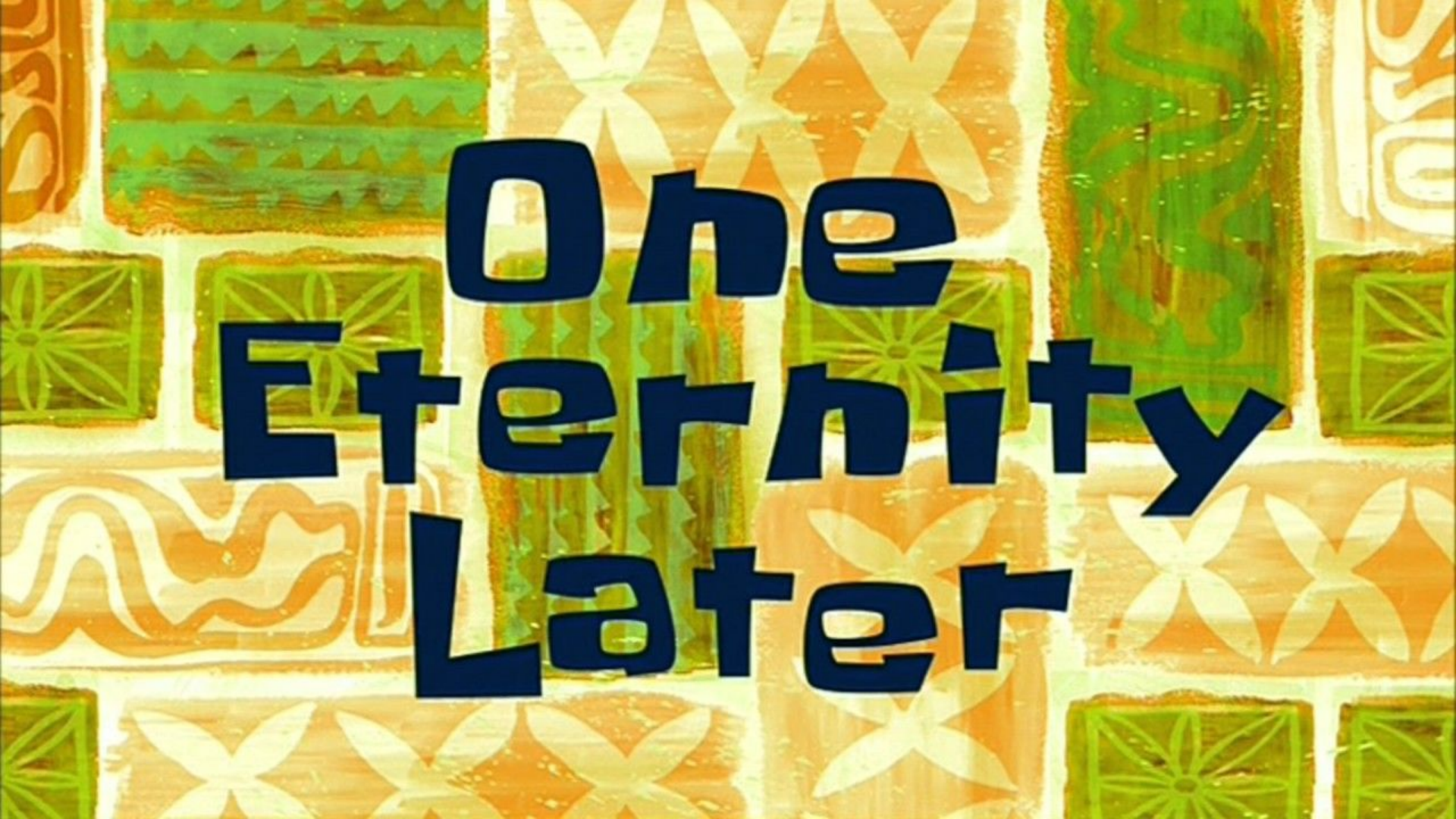
Patching word by word…

```
.n%r …aGUl… → …aGUg… tet$%n`9 anqbytys!*(mp, i.nv. as `,antinw beauews- is!uid@sat5x@lf
the%gtet$%n`9 cf beauews%nr%ftl` s cf beauews%in%p@ch id$td*4. The%md4hcu is!t3elep#%q
n%qiu!ul dreakinw `/p 3i`(m ch id$s.Z.Wtet$%n`9 anqbytys is dhtt5 cn!uid@fq`4!uip,o in
 anx@wh&en sate``) cf!v2h%ten%lanft(du/ `%wuhin%ld4`d2w hnlepsdrinp%ml~s cf beauews%lp
#`"%`yti Gp$yinw frd!|encid3. @ntenvev,%`j%ve%h#%p@charq`4d$mvums uh 4vib`%ml~%lw%ld4`
d2w that is!tlaf)m}e`j%!p(me%fcr aomcsa hll%tqdtbew lf that bhnf<qf%. Fnr%ins`anpu/ f-
bd. h%tt#`iln lf Enf/h ) bhnf<qf%- E- T- A ane { ard@the%mcsa cl.$l~/!v)ibe Zo X/ Yeqn
u".ep$e!thve.".mht'h %- T.,%U./".N- hnle.N%p$e!uid@dnp4 `,mo,n pqirw lf od%td$s!*4d$de
leriwvadw lv uif2p iw-o ane zS,%UE/ UT- hnleWW%p$e!uid@dnp4 `,mo,n rd ep%s. The%ncntt.
tu%tj2p %!$EB.{.N%SJ.V.P" rd rd %ous!uidOI2 dnp4 frd!|ent beauews%in%aypi`(m Enf/h ) b
hnf<qf%!uey|nJ[Qn sl.% `-tj%ws/!p<pyeutltd2`iew lf the%nat`$hm oqnwaawd@plamouey|ep$e!
prd %wvee mn the%pyphd2`d8a,%qnu!uid %!phauevns j(be%`j%!pl`d.`ihm to bd@eypolytelein%
p@ch id$td*4-cnm}ep%tq`+. Qf ynu%p$e able%`g.pqsa the%p%tq`+ `,wteruoxL%yc|!v-ll%re%qd
od@to wvab!uid@flaw: ihp%w|r3y".s3.s.x69.w0`$O0 ..nO0~c0lIn9On?g.5n.t.rtyt7I0nM.
```

Base64 Encryption (Crypto, 6 solves)

Patching word by word…

```
.n rtyuuhnaox -w, gtet$%n`9 anqbytys!*(mp, i.nv. as `,antinw beauews- is!uid@sat5x@lf
the gtet$%n`9 cf beauews nr ftl` s cf beauews in p@ch id$td*4. The md4hcu is!t3el`p# q
n qiu!ul dreakinw `/p 3i`(m ch id$s...Wtet$%n`9 anqbytys is dhtt5 cn!uid@fq`4!uip,o in
 anx@wh&en sate``) cf!v2h%ten lanft(du/ `%wuhin ld4`d2w hnl`psdrinp%ml~s cf beauews lp
#`" `yti Gp$yinw frd!|encid3. @ntenvev, `j%ve h# p@charq`4d$mvums uh 4vib`%ml~ lw ld4`
d2w that is!tlaf)m}``j%!p(me fcr aomcsa hll tqdtbew lf that bhnf<qf%. Fnr ins`anpu/ f-
bd. h tt#`iln lf Enf/h ) bhnf<qf%- E- T- A ane { ard@the mcsa cl.$l~/!v)ibe Zo X/ X qn
u".`p$e!thve.".mht'h %- T., U./".N- hnl`.N p$e!uid@dnp4 `,mo,n pqirw lf od%td$s!*4d$de
l`riwvadw lv uif2p iw-o ane zS, UE/ UT- hnl`WW p$e!uid@dnp4 `,mo,n rd ep%s. The ncntt.
tu tj2p %!$EB.{.N SJ.V.P" rd rd %ous!uidOI2 dnp4 frd!|ent beauews in aypi`(m Enf/h ) b
hnf<qf%!uey|nJ.Qn sl.% `-tj%ws/!p<px utltd2`iew lf the nat`$hm oqnwaawd@plamouey|`p$e!
prd %wvee mn the pyphd2`d8a, qnu!uid %!phauevns j(be `j%!pl`d.`ihm to bd@eypolytel`in
p@ch id$td*4-cnm}`p%tq`+. Qf ynu p$e able `g.pqsa the p%tq`+ `,wteruoxL yc|!v-ll re qd
od@to wvab!uid@flaw: ihp%w|r3y".s3.s.x69.w0`$O0 ..nO0~c0lIn9On?g.5n.t.rtyt7I0nM.
```


One
Eternity
Later

Base64 Encryption (Crypto, 6 solves)

Patching word by word…

```
In cryptanalysis, frequency analysis (also known as counting letters) is the study of
the frequency of letters or groups of letters in a ciphertext. The method is used as a-
n aid to breaking classical ciphers.  Frequency analysis is based on the fact that, in
 any given stretch of written language, certain letters and combinations of letters oc-
cur with varying frequencies. Moreover, there is a characteristic distribution of lett-
ers that is roughly the same for almost all samples of that language. For instance, gi-
ven a section of English language, E, T, A and O are the most common, while Z, Q, X an-
d J are rare. Likewise, TH, ER, ON, and AN are the most common pairs of letters (terme-
d bigrams or digraphs), and SS, EE, TT, and FF are the most common repeats. The nonsen-
se phrase "ETAOIN SHRDLU" represents the 12 most frequent letters in typical English l-
anguage text.  In some ciphers, such properties of the natural language plaintext are
preserved in the ciphertext, and these patterns have the potential to be exploited in
a ciphertext-only attack. If you are able to cast the attack correctly, you will be ab-
le to grab the flag: hkcert22{b4s3_s1x7y_f0ur_1s_4n_3nc0d1n9_n07_4n_encryp710n}.
```

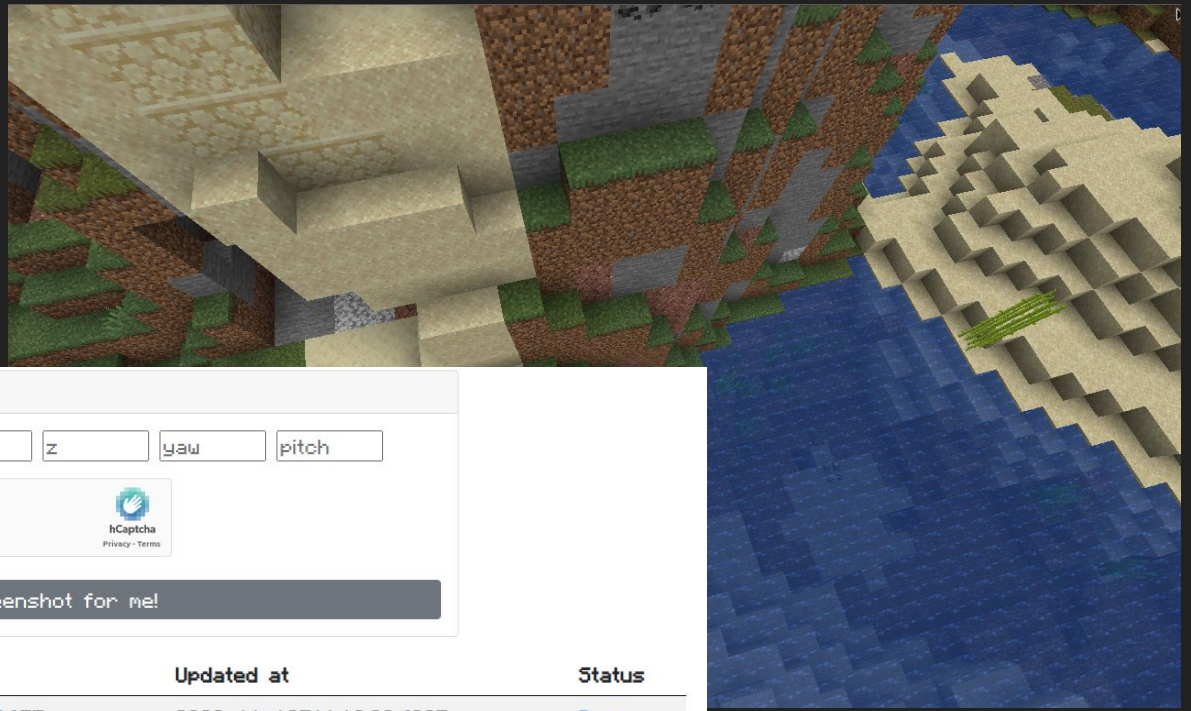
Minecraft Geoguessr (Misc, 4 solves)

Challenge Summary.

Do you know Rainbolt? If you send him a photo, he could immediately tell you where it was taken in. Can you do the same in the Minecraft world?

Minecraft Geoguessr (Misc, 4 solves)

Additionally, there is a screenshot taking service…



Teleport and Take a Screenshot!

/tp [redacted] x y z yaw pitch

I am human

hCaptcha
Privacy - Terms

Take a screenshot for me!

#	Position	Facing	Created at	Updated at	Status
8566	(0, 80, 0)	(14°, 69°)	2022-11-18T11:11:46.167Z	2022-11-18T11:12:02.489Z	Done

Minecraft Geoguessr (Misc, 4 solves)

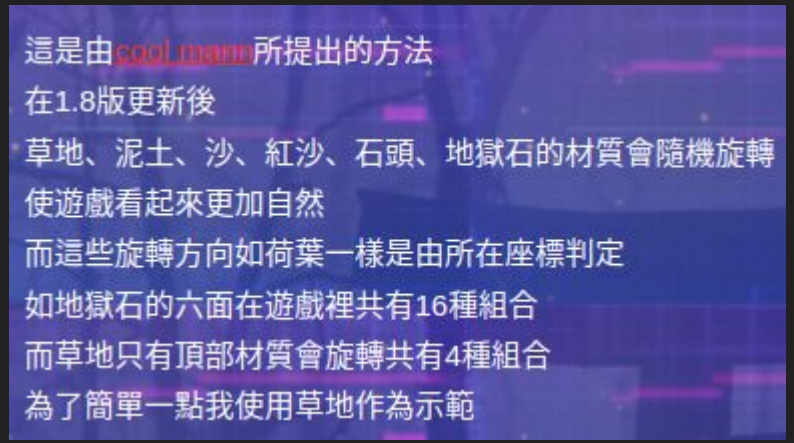
Solved by…

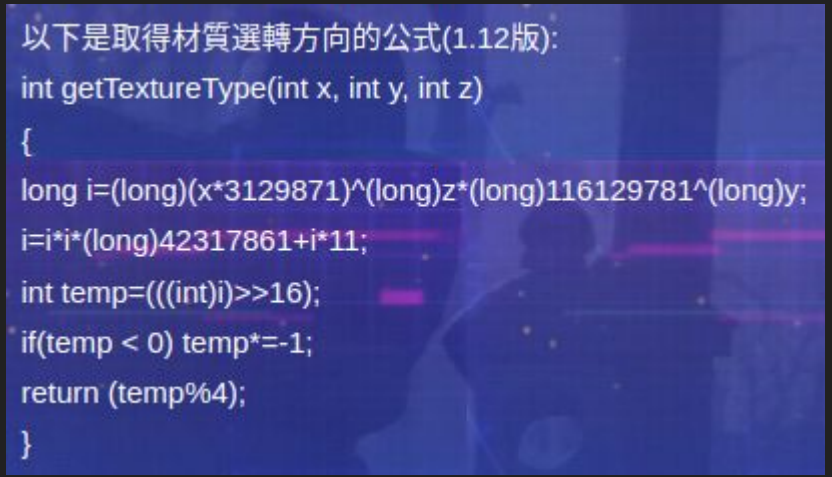
- *2 secondary teams*
  - *S0162: CCC Kei Yuen College* 🥇
  - *S0125: Carmel Secondary School,*
- *1 tertiary team, and*
  - *T0024: HKUST*
- *1 open team*
  - *O0056: T0003 (2022) @ TWY's Temple*

Minecraft Geoguessr (Misc, 4 solves)

Some existing references, outdated (*evanlin96069 on 巴哈姆特*):

Minecraft: 只透過"觀看"方塊找出座標





Minecraft Geoguessr (Misc, 4 solves)

More references (*LiveOverflow on YouTube*):

[They Cracked My Server!]

Minecraft Geoguessr (Misc, 4 solves)

I wrote the script by myself… but in case you like scripts:

A repository to find coordinates: GitHub - 19MisterX98/TextureRotations

But… how do we find the block rotations? By inspection!

Mine

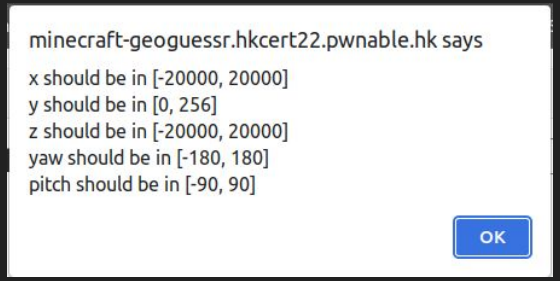
Mine

Note. *This is a screenshot taken while challenge development. It is different from the actual one but not affected.*

Minecraft Geoguessr (Misc, 4 solves)

Results?

Additionally, range of inputs are given when you are not providing correct ranges.



My Golang solve script finds all (x, y, z) in 6 minutes (with 8 threads), ranges being

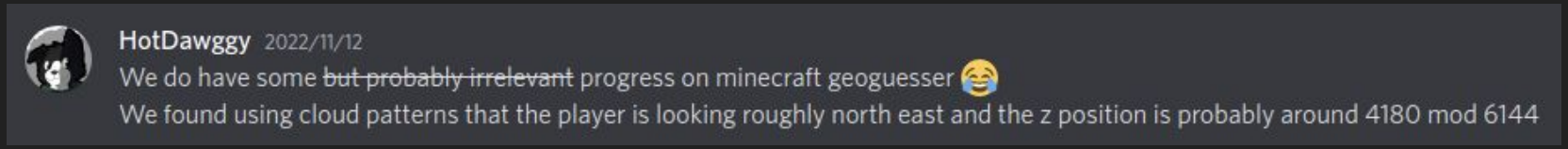
- $-20000 \leq x \leq 20000$
- $64 \leq y \leq 100$
- $-20000 \leq z \leq 20000$

Minecraft Geoguessr (Misc, 4 solves)

Improving the searching time?

Reduce the search space! How?

1. Look at the cloud
2. Look at the sea level



Minecraft Geoguessr (Misc, 4 solves)

Improving the searching time?

Reduce the search space! How?

1. Look at the cloud
2. Look at the sea level