I2P - How can you create a Base32 address from a Base64?

In I2P there are Base64 addresses which point directly from the target. You cannot enter this directly in the address bar. You need it, for example, to connect to an eepsite if you use the BOB or SAM API. Base32 addresses are shorter. They end with ".b32.i2p" and you can enter them in the address bar. If you use the BOB or SAM API, you first have to resolve them to a Base64 address. The local address book is not used for this process, but the I2P network. Therefore, the resolution will only be successful if the corresponding lease set of the eepsite can be found, i.e. if the eepsite's I2P router is online.[1]

How can you convert a Base64 to a Base32 address?

First you decode the Base64 and save it. Please note that the Base64 from I2P does not use the standard decoding alphabet. This would be the following:
`ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/`
and for padding = is used. The decoding alphabet for I2P looks similar:
`ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-~`
and here too = is used for padding. Depending on the program or programming language, there are two ways to decode the base64:

1. Either you use the I2P decoding alphabet and then decode the base64
2. or you replace the `-` with + and the ~ with `/` in the Base64, so that you can decode it with the standard decoding alphabet.

Then take the first 391 bytes of the decoded base64 and take the SHA256 sum. It may also be that the decoded base64 is shorter than 391 bytes. In this case you simply take the SHA256 sum.

After doing this, code the SHA256 sum to a Base32 string. For the Base32 coding, the standard decoding alphabet for the base32 is used.

Next, remove all = signs from base32 and append `.b32.i2p`. Now you have the Base32 address.

Example

A base64 is for example the following:

```
Okd5sN9hFWx-
sr0HH8EFaxkeIMi6PC5eGTcjM1KB7uQ0ffCUJ2nVKzcsKZFHQc7pLONjOs2LmG5H-
2SheVH504EfLZnoB7vxoamhOMENnDABkIRGGoRisc5AcJXQ759LraLRdiGSR0WTHQ0
O1TU0hAz7vAv3SOaDp9OwNDr9u902qFzzTKjUTG5vMTayjTkLo2kOwi6NVchDeEj9M
7mjj5ySgySbD48QpzBgcqw1R27oIoHQmjgbtbmV2sBL-2Tpyh3lRe1Vip0-K0Sf4D-
Zv78MzSh8ibdxNcZACmZiVODpgMj2ejWJHxAEz41RsfBpazPV0d38Mfg4wzaS95R5h
Hx6eh7oG61KBN1cO1oY8M2oQfISCCpWHPm4WzOlfOHyZOzUHOpcYERAFbsO~w0-
```

1 https://geti2p.net/en/docs/naming#base32

`ryKIjZJeTzmc314vQB3gG8zMhPRWy9ff6BMkI492DQG2Qv0y3f5fRtx3BfzD7POhuj`
`oICzbs8XiCOuuAu8BO0u3k903uHAPNurZx~0jG05TdBQAEAAcAAA==`
This is now decoded to generate the Base32 address and then has a length of 391 bytes. So you don't have to do anything anymore and can take the SHA256 sum. In hex format this would look like this:
`5430f325e9b45e76e48170fa4aee72d56684789d9b6713722d2a13017e387ac7`
However, one does not take the SHA256 sum in hex format, but in bytes the Base32 string and receives the following:
`kqypgjpjwrphnzebod5ev3ts2vtii6e5tntrg4rnfijqc7rypldq====`
From this Base32 we now cut away the = sign and append `.b32.i2p`. Then you get the following:
`kqypgjpjwrphnzebod5ev3ts2vtii6e5tntrg4rnfijqc7rypldq.b32.i2p`

Note and notices

To generate the Base32 address, you take the SHA256 sum. This is also the reason why it is extremely difficult or even impossible to create a Base64 from a Base32. Furthermore, some prefer to use Base32 address instead of Eepsite names. This is because an eepsite name is resolved with the help of the address book and the Base64 has been maliciously changed. However, the Base32 address is resolved every time by the I2P network. Therefore you cannot maliciously change the Base64 there. Especially with inproxies, where you can edit the address book, you should make sure to use Base32 addresses.

Practical use

There are several scripts that transform a base64 into a base32:

- https://github.com/i2p/i2p.scripts/blob/master/b32/64to32.sh This script is in the public domain. It was written by HungryHobo and is an SH script.
- https://github.com/eyedeekay/keyto This script was written by I2P developer idk in Go.
- https://github.com/eyedeekay/i2pasta/blob/master/convert/64232.go Alternatively there is this script, which was also written by idk in Go. It should be noted that the scripts work pretty much the same.
- https://test.mk16.de/scriptFiles/i2pb64tob32.rb This script was written by Marek in Ruby. It uses an external library for Base32 coding, which is, however, in the script. The library is available under the MIT license. The rest of the code under the WTFPL license.

I2P – Wie kann man aus einer Base64 eine Base32-Adresse erzeugen?

In I2P gibt es Base64 Adressen, welche direkt aus das Ziel verweisen. Diese kann man nicht direkt in die Adressleiste eingeben. Man braucht sie aber zum Beispiel zum Verbindung zu einer Eepsite, wenn man das BOB oder SAM API

benutzt. Base32 Adressen sind kürzer. Sie enden mit „.b32.i2p“ und man kann die in die Adressleiste eingeben. Wenn man das BOB oder SAM API verwendet, muss man sie aber erst in eine Base64 Adresse auflösen. Für diesen Vorgang wird nicht das lokale Adressbuch benutzt, sondern das I2P-Netzwerk. Daher wird die Auflösung auch nur erfolgreich sein, wenn das entsprechende LeaseSet der Eepsite gefunden werden kann, also wenn der I2P-Router der Eepsite online ist.[2]

Wie kann man nun eine Base64 in eine Base32-Adresse konvertieren?

Als erstes dekodiert man die Base64 und speichert diese. Dabei ist zu beachten, dass die Base64 von I2P nicht das Standarddekodierungsalphabet benutzt. Dieses wäre folgendes:
`ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/`
und für das Padding wird = benutzt. Das Dekodierungsalphabet für I2P sieht ähnlich aus:
`ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-~`
und auch hier wird für das Padding = benutzt. Es gibt also jenach Programm bzw. Programmiersprache zwei Möglichkeiten die base64 zu dekodieren:

1. Entweder benutzt man das I2P Dekodierungsalphabet und dekodiert dann die base64
2. oder man ersetzt in der Base64 die `-` mit `+` und die `~` mit `/`, sodass man es mit dem Standarddekodierungsalphabet dekodieren kann.

Danach nimmt man die ersten 391 Bytes der dekodierten base64 und nimmt davon die SHA256 Summe. Es kann auch sein, dass die dekodierte base64 kürzer als 391 Bytes ist. In diesem Fall nimmt man dann einfach davon die SHA256 Summe.

Nach dem man das getan hat, codiert man die SHA256 Summe zu einem Base32 String. Für die Base32 Codierung wird das Standarddekodierungsalphabet für die base32 genommen.

Als nächstes entfernt man alle =-Zeichen von der base32 und hängt `.b32.i2p` an. Nun hat man die Base32-Adresse.

Beispiel

Eine base64 ist zum Beispiel folgende:

```
Okd5sN9hFWx-
sr0HH8EFaxkeIMi6PC5eGTcjM1KB7uQ0ffCUJ2nVKzcsKZFHQc7pLONjOs2LmG5H-
2SheVH504EfLZnoB7vxoamhOMENnDABkIRGGoRisc5AcJXQ759LraLRdiGSR0WTHQ0
O1TU0hAz7vAv3SOaDp9OwNDr9u902qFzzTKjUTG5vMTayjTkLo2kOwi6NVchDeEj9M
7mjj5ySgySbD48QpzBgcqw1R27oIoHQmjgbtbmV2sBL-2Tpyh3lRe1Vip0-K0Sf4D-
Zv78MzSh8ibdxNcZACmZiVODpgMj2ejWJHxAEz41RsfBpazPV0d38Mfg4wzaS95R5h
```

2 https://geti2p.net/en/docs/naming#base32

`Hx6eh7oG61KBN1cO1oY8M2oQfISCCpWHPm4WzOlfOHyZOzUHOpcYERAFbsO~w0-ryKIjZJeTzmc314vQB3gG8zMhPRWy9ff6BMkI492DQG2Qv0y3f5fRtx3BfzD7POhujoICzbs8XiCOuuAu8BO0u3k903uHAPNurZx~0jG05TdBQAEAAcAAA==`
Diese wird nun um die Base32-Adresse zu erzeugen dekodiert und hat danach eine Länge von 391 Bytes. Also muss man nichts mehr tun und kann die SHA256 Summe nehmen. In Hex-Format würde diese so aussehen:
`5430f325e9b45e76e48170fa4aee72d56684789d9b6713722d2a13017e387ac7`
Man nimmt allerdings nicht von der SHA256 Summe in Hex-Format, sondern in Bytes den Base32 String und erhält folgendes:
`kqypgjpjwrphnzebod5ev3ts2vtii6e5tntrg4rnfijqc7rypldq====`
Von dieser Base32 schneiden wir nun die =-Zeichen weg und hängen `.b32.i2p` an. Dann erhält man folgendes:
`kqypgjpjwrphnzebod5ev3ts2vtii6e5tntrg4rnfijqc7rypldq.b32.i2p`

Anmerkung und Hinweise

Um die Base32-Adresse zu erzeugen, nimmt man also die SHA256 Summe. Dies ist auch der Grund, warum es extrem schwer oder sogar unmöglich ist, aus einer Base32 eine Base64 zu erzeugen. Des Weiteren bevorzugen manche die Verwendung von Base32-Adresse anstatt von Eepsite-Namen. Dies liegt daran, dass ein Eepsite-Name mit Hilfe des Adressbuches aufgelöst wird und so die Base64 bösartig verändert wurde. Die Base32-Adresse wird allerdings jedes mal durch das I2P-Netzwerk aufgelöst. Daher kann man die Base64 dort nicht bösartig verändern. Besonders bei Inproxies, bei welchen man das Adressbuch bearbeiten kann, sollte man darauf achten am besten Base32-Adressen zu verwenden.

Praktische Anwendung

Es gibt mehrere Skripts, welche eine Base64 in eine Base32 verwandelt:

- https://github.com/i2p/i2p.scripts/blob/master/b32/64to32.sh Dieses Skript ist gemeinfrei (also Public Domain). Es wurde von HungryHobo geschrieben und ist ein SH-Skript.
- https://github.com/eyedeekay/keyto Dieses Skript wurde vom I2P-Entwickler idk in Go geschrieben.
- https://github.com/eyedeekay/i2pasta/blob/master/convert/64232.go Alternativ gibt es dieses Skript, welches auch von idk in Go geschrieben wurde ist. Dabei sollte man beachten, dass die Skripts recht gleich funktionieren.
- https://test.mk16.de/scriptFiles/i2pb64tob32.rb Dieses Skript wurde von Marek in Ruby geschrieben. Es verwendet für die Base32-Kodierung eine externe Bibliothek, welche allerdings im Skript ist. Die Bibliothek steht unter der MIT Lizenz zur Verfügung. Der Rest des Codes unter der WTFPL Lizenz.