Mapping public keys into a colored circle

Problem: encode (part of) a public key given in hex in a colored circle via a unique mapping.

Idea:

- assume the circle has length N=25 and we have C=32 colors used for the $K \in \{2, ...N-1\}$ segments of the circle $\rightarrow$ we can simply calculate all possible, colored circles that can be generated
- We store these in a dictionary, then we map (part of) the public key onto a integer (for example, from hex to int). This integer gives the unique circle in the dictionary.
- By constructing this dictionary and keeping the algorithm general (N, K, C), we also know the exact number of possible colored circles and in addition can easily introduce further constraints or enlarge the set.
- The algorithm needs to do two things:
  1. Give a set that contains all relevant partitions of the circle.
  2. Give a coloring to the set.

Parameter analysis of a secure map of status.im public keys

The purpose of the this document to provide a brief analysis of given options to securely map status.im 512 bit chat keys/public keys to a different encoding, leveraging a combination of emojis and colors (and optionally additional factors) to encode the key.

Mapping public keys into two alphabets

Given a key of 512 bit-length, we can map this key to a different representation:

$$M : [0,1]^{512} \rightarrow [0,1024]^n \times [0,32]^k \tag{1}$$

with n and k integers.

To ensure the mapping doesn't break the security of the overall scheme, it needs to be bijective. We need to choose n and k accordingly:

$$2^{512} = 1024^n \cdot 512^k \tag{2}$$
$$2^{512} = (2^{10})^n \cdot (2^5)^k \tag{3}$$
$$2^{512} = 2^{10 \cdot n + 5 \cdot k} \tag{4}$$

and taking log_2 on both sides gives us

$$512 = 10 \cdot n + 5 \cdot k \tag{5}$$

Whereas in bit-representation, our key has 512 characters, this number is reduced by the mapping M. If we map it to 0x, we need 128 chars. The challenge is to define a mapping that yields significant compression. If we only take one representation on the right hand side, we reduce the representation to roughly 52 for k=0 (only emojis) or 103 for n=0 (only colors).

Combining both representations, we get a trade-off between the large one and the smaller one. In our case, this is

$$n = \frac{512}{10} - \frac{k}{2} \tag{6}$$

hence, **increasing k by 2 leads to a reduction in n by only 1.**

To illustrate this, let us start from $k = 0$ (colors). Then we need $n = 52$ (emojis). Increasing now k to 10, we still need roughly $n = 47$. For $k = 20$, we need $n = 42$ and for $k = 30$ we get $n = 37$.

Note that since n and k need to be integers, we sometimes have to round them up to ensure that our mapping can be bijective.