Offline encryption of seed recovery phrases

Published by: Wladimir Weinbender, founder of CRYPTOETCH (https://cryptoetch.com/)
Date: 19 September 2019

Everyone who has accumulated a small fortune in crypto will inevitably start caring about security. In crypto, security is all about safely storing the seed recovery phrase (usually a list of 12 or 24 English words). The seed recovery phrase allows you to recover your crypto funds in case your hardware wallet device gets lost or damaged. As anyone else who gets access to your seed phrase will be able to access and steal your crypto funds, you should keep it safe like a treasure.

```
witch collapse practice feed shame open despair creek road again ice least
```

Example of a 12-word seed recovery phrase

Most people store their seed phrases as plaintext (i.e. readable to anyone) on a piece of paper or on a metal plate. This is very risky though. What if your loved ones take a picture of your phrase and put it online? What if you lose your phrase or it gets stolen? In any case, you can assume that your crypto funds will be gone.

In this article, I am going to describe a method to turn your seed phrase into an unreadable piece of text by means of a code (also referred to as “cipher”) just using pen and paper. Even if someone discovers your obscured phrase, your cryptocurrency is unlikely to be stolen. The presented method is not exclusive to seed phrases, but can be applied to any sort of secret information that you want to hide (e.g., passwords).

Caesar already did it

The process of obscuring secret information by means of a cipher is known as encryption. The history of encryption can be traced back to the very earliest civilizations. One of the most famous encryption techniques is known as the substitution cipher and was used by Caesar for sending confidential messages to his troops. In a substitution cipher each letter is replaced with a different letter or symbol to produce an unreadable piece of text. Caesar himself used a shift of 3 which means that “A” was encrypted as “D”, “B” as “E”, “Z” as “C” etc. The table below illustrates a fixed shift of 3 applied to the plaintext “ATTACK THE ENEMY”. The resulting encrypted message is also referred to as ”ciphertext”.

Plaintext	A	T	T	A	C	K	T	H	E	E	N	E	M	Y
Shift	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Ciphertext	D	W	W	D	F	N	W	K	H	H	Q	H	P	B

Substitution cipher using a fixed shift of 3

If the shift key is known to the intendent recipient, then the secret message can be easily deciphered by shifting each letter back by 3.

While shifting the letters of a message with a constant key is very simple to implement and to remember, it is not very secure. If it is known that a fixed shift cipher has been used, but the key itself is unknown, then it is simple to break the code by trying to shift back the letters by attempting all 25 possible keys in succession (ignoring the identity mappings 0 and 26) until a plaintext that makes sense is returned. It is unlikely that

two different shift keys will produce a second plaintext that makes sense. It is worth highlighting that the number of trials to break the code is independent of the message length.

Can we make it better?

Yes we can. Instead of shifting each letter with a constant key, we could shift each letter with a different key ranging from 0 to 99. An example follows below (an easy way to shift letters will be presented in a later chapter):

Plaintext	A	T	T	A	C	K	T	H	E	E	N	E	M	Y
Shift	2	7	14	39	3	15	0	9	26	25	41	22	12	11
Ciphertext	C	A	H	N	F	Z	T	Q	E	D	C	A	Y	J

Substitution cipher using variable shift keys

Instead of 25 possible solutions for the full message, we now have 25 possible solutions for one single letter (26 including the identity mapping). For a message of length n, this implies 26^n possible solutions (note that this includes constant shifts and the identity mapping as special cases that we consciously do not adjust for, for simplicity reasons). For our 14 letter example, this implies $26^{14} \sim 6.5\ x\ 10^{19}$ possible solutions. To put this into perspective, this is roughly 8 billion times the current population of the earth (c. 8 billion).

Clearly, there will be many messages that one can directly exclude because they are not meaningful. At the same time, there will be also many 14 letter messages that will be meaningful and could even have contradicting meanings. For instance, “Wait with Attack” could be one possible solution, not only to our specific example but to any 14-letter string:

Ciphertext	J	H	M	G	V	E	V	G	G	Z	A	G	L	K
Shift	13	7	4	13	25	22	2	25	6	6	7	6	9	26
Plaintext	W	A	I	T	W	I	T	H	A	T	T	A	C	K

One possible solution for a 14-letter message

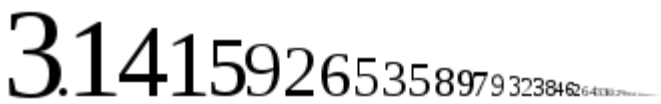
The fact that there is no uniquely identifiable solution discourages a brute force approach in the first place. The only way to decipher the message is to get access to the shift key sequence.

Generating a sequence of random numbers

The question that you might have at this stage is how to remember such a long sequence of numbers. Some people cannot remember their own phone number involving less than 15 digits, how can we expect them to remember a string of more than 100 numbers in the case of a 24-word seed recovery phrases? The answer is to derive the sequence of numbers from an easy-to-remember mathematical constant/expression. Hence, instead of remembering the sequence of numbers itself, you just need to remember the mathematical constant/expression that generated it.

Let me give you an example. Consider the famous mathematical constant $\pi = 3.1415926\ldots$. It has an infinite decimal representation and the digits 0 through 9 appear in a random order with no repeatable pattern (numbers that have these characteristics are referred to as irrational numbers; other common

examples of irrational numbers include $e = 2.7182818 \ldots$, which is the mathematical constant whose natural logarithm is equal to one, or square roots such as $\sqrt{2} = 1.4142135 \ldots$).



Source: Wikipedia. The mathematical constant π is an irrational number with an infinite decimal representation.

Let us define the first decimal digit as the starting point of our shift sequence and let us combine two digits to form a shift key. Using π will give us the following shift cipher and ciphertext:

Plaintext	A	T	T	A	C	K	T	H	E	E	N	E	M	Y
Shift	14	15	92	65	35	89	79	32	38	46	26	43	38	32
Ciphertext	O	I	H	N	L	V	U	N	Q	Y	N	V	Y	E

Substitution cipher derived from the mathematical constant π

Note that the only thing you need to remember is that you used π to derive the shift sequence. It's like a PIN that comes in the form of a mathematical constant/expression. There is no need to remember the sequence itself anymore. Isn't that great?

π, e, and $\sqrt{2}$ are very simple choices. While you could certainly use them to encrypt your data, an attacker who has read this article might try them out first. Hence, for security reasons, I recommend to be more creative and include standard mathematical operations such as addition, subtraction, multiplication, division, exponentiation and other well-known functions such as square root, sin/cos/tan, or logarithm to come up with your mathematical expression.

Five examples of more complex expressions are provided below (use a powerful online calculator like wolframalpha to evaluate mathematical expressions up to a precision of almost 2'000 decimal digits, which is more than enough for our purposes):

Mathematical expression	**Decimal representation**
$(\pi * e)^{3.21}$	977.094071587120282554733902516498593686833816073995219818 6 …
$\pi + 1.1 * e$	6.131702664894742997358959601767431631730141202445061353438 …
$2 * \sqrt{\pi + 1}$	4.070180661145052042055969740923178731889133324298210668940 …
$\ln(\sqrt{2})$	0.346573590279972654708616060729088284037750067180127627060 …
$\sin(2) * \cos(4)$	−0.594356462512303784103787656261819801165124917883047293 43 …

Five examples of mathematical expressions and their decimal expansions

To demonstrate the strength of the presented encryption technique, I am publicly posting an encrypted 12-word seed recovery phrase that comes with 0.0105 BTC (c. $100 at the time of writing). Note that it is sufficient to store the first four letters of each word (three letters for three letter words):

Encrypted seed recovery phrase: ZQWWFMWFWVDNTIWTBCNXAZNMIHBBRRBPDOLHPQHAWZNIRUBF

You can verify the ownership of the address by following the instructions here using below information.

- Message: I love CRYPTOETCH
- Public address: bc1qqd7sxycp3vv7tf2efssa3yktfgpqjapxjankxh

- Signature:
  IHDMsTyDUHvOzeNVth4FPDgWVArEhHvoUtgKErQmOmOlQPz9ToLfZHelcXssEubDuHAmxGmRw8Ad7wUdD3ufXyU=

Without knowing the shift cipher, it is impossible to break it. The fact that you can protect your assets even in the case of loss or theft by means of simple encryption is just another example why digital assets like bitcoin are superior to physical assets like gold or silver.

Shifting made easy

The beauty of the substitution technique is that the recovery seed phrase can be encrypted without typing it into a computer or phone. Especially when dealing with crypto, this is critical to avoid hacks. While performing the shift mentally is possible, it might be tedious and prone to errors (especially for larger numbers). To facilitate the shift operation, I recommend using the crypto wheel provided by CRYPTOETCH (download the pdf here and print it out on paper). It consists of two discs, one representing the outer circle with letters A-Z and one representing the inner circle with numbers 0-99. The substitution letter can be found by simple rotation. For example, to shift letter “G” by 85, navigate the zero (red colored field) to “G” and read the letter that 85 points to (in this example “N”). To decrypt an encrypted letter using the crypto wheel, just reverse the process. That is, point the shift key value, say 85, to the encrypted letter, say “N”, by rotation; the letter that the red zero field is pointing to (“G” in our case) is the decrypted letter. Two examples are illustrated below.

Two examples using the crypto wheel provided by CRYPTOETCH to facilitate the shift operation.

Discussion

In our example above, we have arbitrarily defined the first decimal digit as the starting point of our shift sequence. It does not really matter where you start your sequence. You can start it at the tenth or the twentieth decimal place. The only thing that matters is that you remember it. I recommend to not overcomplicate things and use the most natural choices which are either the first number or the first decimal number.

Further, we also made a choice on how many digits to combine to form a shift key. We could have chosen each single digit to represent a shift key. This, however, would restrict the key range to 0-9, meaning that any letter could only be replaced by 9 out of 25 possible letters (ignoring the identity mapping). For instance, “A” could only be shifted to “B”, “C”, up to “J”, but never to “K”, “P”, “Z”, or “D”. If an attacker has the information that a key range of 0-9 was used, this can significantly narrow down the set of possible solutions and make your secret piece of information more likely to be uncovered.

I personally recommend combining two digits which extends the key range to 00-99 (where 00 is equivalent to 0, 01 equivalent to 1, 02 equivalent to 2, etc.). This implies that any letter can be replaced by any other letter in the alphabet. While you could certainly choose to combine three or more digits to form a shift key, it will not make your encryption more secure. I suggest to keep it simple and go for two digits.

Last but not least, I suggested to use an online calculator to evaluate mathematic expressions. If you do that, I strongly recommend you to clear the browser history afterwards to erase any traces. If you are still concerned about exposing your mathematical expression to the internet, you could also use a normal calculator, which usually has a precision of 10-15 digits, and build the shift sequence by repetition of the available decimal expansion.

Generating a shift sequence using a mathematical constant/expression is just one possibility. You could also derive it from a phrase taken out of your favorite book. For this, you would convert each letter into a number and use it as your shift cipher. For instance, “HELLO” would translate into 8, 5, 12, 12, 20.

You could take the whole approach one step further and add another layer of security by obfuscating the encrypted message one more time using a permutation cipher. At the end, it’s up to you how you generate your shift sequence and how many layers of obfuscation you apply. The only thing that matters is that you remember how it was derived.

One important advice in terms of storage: never store your encrypted seed recovery phrase together with the expression that generated the shift cipher. Keep them in two separate places.

Conclusion

Many people are hesitant to adopt crypto because of their fears associated with managing their own funds. They are not comfortable with the risk of losing their funds due to unsecure handling of their secret keys. Taking away those fears by providing crypto holders with tools and methods that will help them to store their secret keys more safely is critical for mainstream adoption of crypto. The technique to obfuscate your seed recovery phrases in order to protect your funds in case of loss or theft of your secret keys presented in this article is my contribution to crypto mainstream adoption.

If you liked this article, you can support me by ordering a CRYPTOETCH kit in my online shop (https://cryptoetch.com/) that will provide you with all the ingredients that you need in order to “burn” your encrypted seed recovery phrases into steel by means of a chemical reaction known as “etching”. It is the cheapest way to bring your seed recovery phrases on metal and protect it from water and fire. Every cent of net revenue will be converted back into crypto.

I wish you happy HODLing.