Scientific and Practical Cyber Security Journal (SPCSJ) 6(4): 57-63 ISSN 2587- 4667 Scientific Cyber Security Association (SCSA)

USAGE OF COLLATZ CONJECTURE IN CRYPTOGRAPHY

Andria Kelekhsaevi, Cervantes Gymnasium-gess
Gega Shavdatuashvili, Cervantes Gymnasium-gess
Giorgi Meliqidze, 176 pubic school
Giorgi Mchedlidze, school Opiza

ABSTRACT: This article distinguishes Usage of Collatz Conjecture in cryptography. In particular, how we can hash information using the "3n+1" algorithm. We review a new method of irreversible hash and its usage in the modern world using,"3x+1" problem.

KEYWORDS: *Collatz Conjecture, Cryptography, Collatz problem, irreversible hash, Encryption*

Introduction:

Hashing is the process of transforming any given key or a string of characters into another value. This is usually represented by a shorter, fixed-length value or key that represents and makes it easier to find or employ the original string.

The most popular use for hashing is the implementation of hash tables. A hash table stores key and value pairs in a list that is accessible through its index. Because key and value pairs are unlimited, the hash function will map the keys to the table size. A hash value then becomes the index for a specific element. A hash function generates new values according to a mathematical hashing algorithm, known as a hash value or simply a hash. To prevent the conversion of hash back into the original key, a good hash always uses a one-way hashing algorithm.

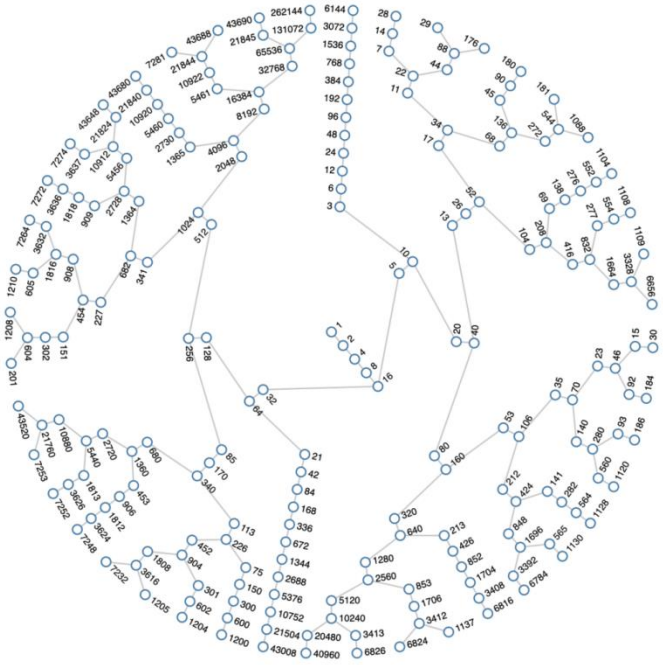
Hashing is relevant to -- but not limited to -- data indexing and retrieval, digital signatures, cybersecurity and cryptography.

Collatz Conjecture:

The Collatz conjecture is one of the most famous unsolved problems in mathematics. The conjecture asks whether repeating two simple arithmetic operations will eventually transform every positive integer into 1**.**

The Collatz conjecture states that the orbit of every number under f eventually reaches 1. And while no one has proved the conjecture, it has been verified for every number less than 268. So if you're looking for a counterexample, you can start at around 300 quintillion [1].

It's easy to verify that the Collatz conjecture is true for any particular number: Just compute the orbit until you arrive at 1.



The importance of Collatz Conjecture in cryptography

By implementing the Collatz conjecture in cryptography we create an irreversible hash. Unlike encryption, Cryptographic Hash Functions are one-way. Once encrypted, you can never decrypt them even if you have the exact hashing algorithm that was used for the encryption [2].

Our Encryption works in that way:

1) We divide the alphabet into 13 pairs and mark each letter with ones and zeros.
The default corresponding number of each letter is zero and it becomes one only when the letter appears in the word [3].

(For example we turn word "hello" in given binary table)

Pairs of letters:		Binary Values:	
a	***b***	*0*	*0*
c	***d***	*0*	*0*
e	***f***	***1***	*0*
g	***h***	*0*	***1***
i	***j***	*0*	*0*
k	***l***	*0*	***1***
m	***n***	*0*	*0*
o	***p***	***1***	*0*
q	***r***	*0*	*0*
s	***t***	*0*	*0*
u	***v***	*0*	*0*
w	***x***	*0*	*0*
y	***z***	*0*	*0*

2) *After the first step of encryption, we add a special number to every sum of ones that correspond to a letter that was used in a given word. We do not add a special number to pairs where one doesn't appear. Special numbers are the same every time and are generated by adding the index of each letter in the same pair [4]. You can view it here:*

Final Value:		
3	*0*	*3*
7	*0*	*7*
11	***+1***	***12***
15	**+1**	**16**
19	*0*	*19*
23	**+1 +1**	***25***
27	*0*	*27*
31	**+1**	***32***
35	*0*	*35*
39	*0*	*39*
43	*0*	*43*
47	*0*	*47*
51	*0*	*51*

3) *After this step of encryption, we are left with the following result:* ***00121602503200000***

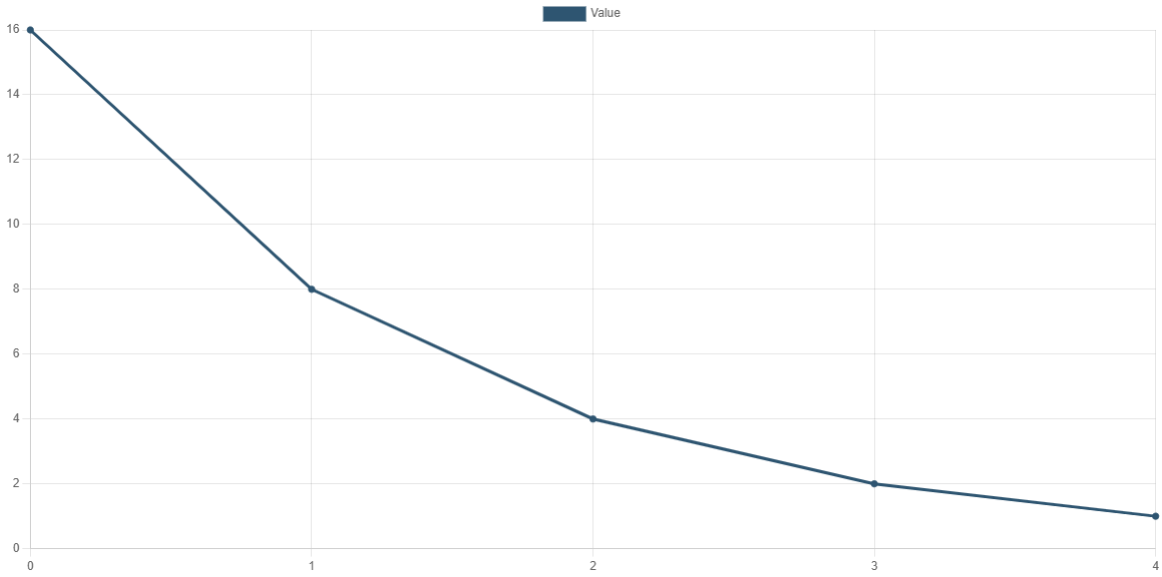
So on step 1 enc(k)("hello") = **00121602503200000**

After turning our string into a number hash we are going to use the 3x+1 problem to hash it even better and make it irreversible. The 3x+1 problem or so-called Collatz problem allows us to create a hash that will be impossible to decrypt because of its random pattern.

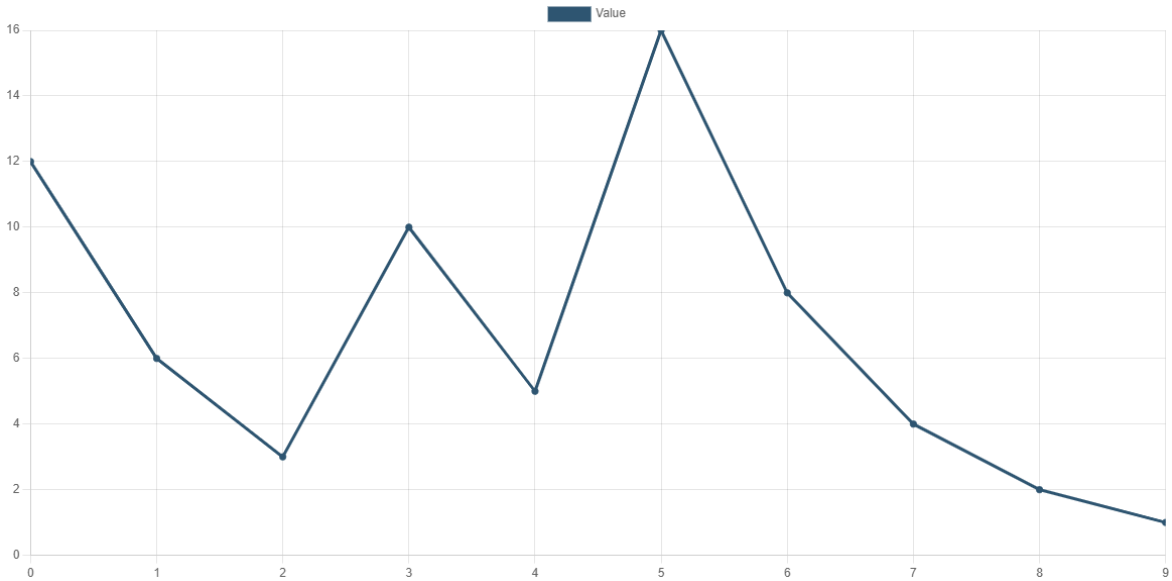
From the above hash, we ignore zeros and put every other number we got into a Collatz function. That means we will use the 3x+1 algorithm on 12,16,25 and 32. After finishing the algorithm we take the highest number in the Collatz tree and correspond it to the number that was used. If two different numbers

have the same highest number in the algorithm then we just choose 2-nd highest (except starting number) and so on [5,6].

Now, let's watch each step:

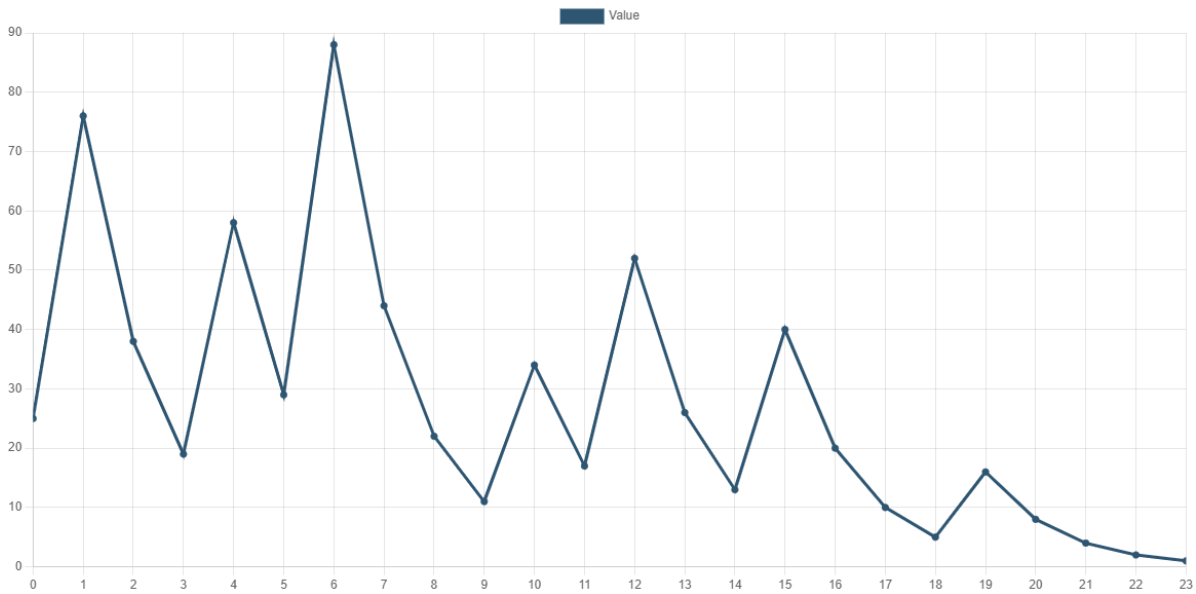


16 -->8 -->4 -->2 -->1
The highest number in this algorithm is 16 itself.
So enc(k)("h") = enc(k)(16) = 16

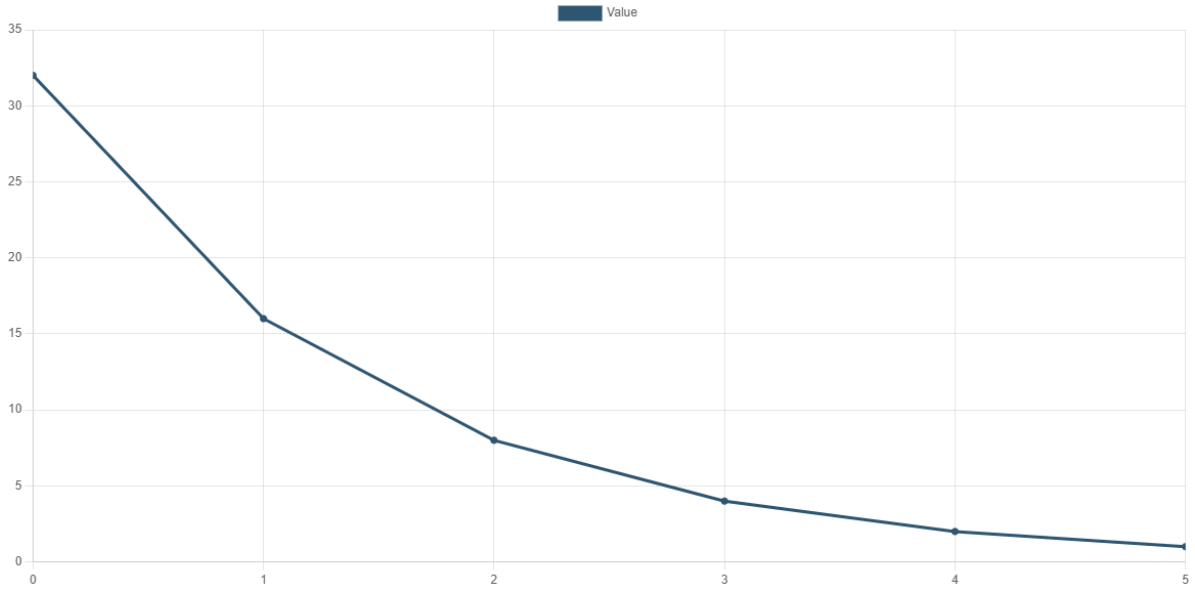


12 -->6 -->3 -->10 -->5 -->16 -->8 -->4 -->2 -->1

The highest number in this algorithm is also 16 so we choose the 2nd highest number (except starting number),which is 10.
soenc(k)("e") = enc(k)(12) = 10



25 -->76 -->38 -->19 -->58 -->29 -->88 -->44 -->22 -->11 -->34 -->17 -->52 -->26 -->13 -->40 -->20 -->10 -->5 -->16 -->8 -->4 -->2 -->1
The highest number in this algorithm is 88 so enc(k)("l") = enc(k)(25) = 88. Since "l" is used twice in hello, we put 88 twice in a final hash.



32 -->16 -->8 -->4 -->2 -->1
Highest number in this algorithm is 32 so enc(k)("o") = enc(k)(32) = 32.

After finishing our last step we are left with a hash that looks like this: **1610888832** so enc(k)("hello") = enc(k)(00121602503200000) = **1610888832**
After the whole hashing process word "hello" turned into a 1610888832. We would like to prove why it's impossible to decrypt a hash generated by our algorithm. Even if the attacker knows the method that was used they won't be able to decrypt the hash because of a simple reason. The Collatz problem. They won't be able to find the exact
number that was used as a starting point in our algorithm. Even If starting number might not change after the algorithm (like it happened with "h" when 16 was = 16) it is still impossible. Let's say the attacker knows that 88 was the highest number in our algorithm [7]. They put 88 in the Collatz conjecture algorithm and are left with the next problem:

88 -->44 -->22 -->11 -->34 -->17 -->52 -->26 -->13 -->40 -->20 -->10 -->5 -->16 -->8 -->4 -->2 -->1

The attacker might think that they will be able to find the number we used to hash but in reality, they have to do the
reverse path search [8]. To make it simple, we used the number 25 to get the number 88, and even if the attacker uses the back road:

reverse:
88 -->29 -->58 -->19 -->38 -->76 -->25
They will keep on going because of a simple reason. The attacker won't know where to stop. Our hashed number was
25 but the attacker will keep on searching for the starting number:
88 -->29 -->58 -->19 -->38 -->76 -->25 -->50 --> and so on.

Conclusion:

In the given article, we overviewed hashing and explained why and how Collatz Conjecture (3x+1 Problem) can be used in cryptography. We discussed an algorithm which sorts the English alphabet in 13 pairs and creates a binary table where each binary number corresponds to a specific letter. After the first stage of hashing our algorithm sums all the ones in the same row and adds a special number to each result. Final step is to put the result into a Collatz conjecture algorithm and pick the highest number in the path.

Over 80 years Collatz problem remains unsolved, so implementing it in cryptography gives a new, unique method of hashing which is irreversible and can’t be decrypted.

Acknowledgement: This work was supported by Shota Rustaveli National Science Foundation of Georgia [SPG-22-218]

BIBLIOGRAPHY

1. L. Colussi, “The convergence classes of Collatz function,” *Theoretical Computer Science*, vol. 412, no. 39, pp. 5409–5419, 2011.
2. P. C. Hew, “Working in binary protects the repetends of 1/3h: Comment on Colussi’s ’The convergence classes of Collatz function’,” *Theoretical Computer Science*, vol. 618, pp. 135–141, 2016.
3. R. K. Guy, “Don’t try to solve these problems,” *Computers and Mathematics with Applications*, vol. 90, no. 1, pp. 35–41, 1983.
4. G. T. Leavens and M. Vermeulen, “ search programs,” *Computers & Mathematics with Applications. An International Journal*, vol. 24, no. 11, pp. 79–99, 1992.
5. R. E. Crandall, “On the "3x+1" problem,” *Mathematics of Computation*, vol. 32, no. 144, pp. 1281–1292, 1978.
6. W. Ren, S. Li, R. Xiao, and W. Bi, “Collatz Conjecture for 2100000-1 is true - algorithms for verifying extremely large numbers,” in *Proceedings of the IEEE UIC 2018*, pp. 411–416, Guangzhou, China, October 2018.
7. I. Krasikov and J. C. Lagarias, “Bounds for the problem using difference inequalities,” *ActaArithmetica*, vol. 109, no. 3, pp. 237–258, 2003. View at: Publisher Site | Google Scholar | MathSciNet
8. Gagnidze, M. Iavich, G. Iashvili, Some Aspects of Post-Quantum Cryptosystems, Abstract book, EURO-ASIA FORUM IN POLITICS ECONOMICS AND BUSINESS – 2016, JULY 21-22, 2016, BELGRADE, SERBIA.