Alpha-Numeric Base Decoder using Python

Amer Bindiyab[1], Mohammed Ishaq[2], Mohammad Rahmat Ali[3]

[12]B.E STUDENT, [3]ASSISTANT PROFESSOR

Department of CSE

ISL Engineering College, Bandlaguda, Hyderabad.

ABSTRACT- Alphanumeric Base Decoder is a tool written in Python that will be able to decode all alphanumeric base encoding schemes. We can Decode Base16, Base32, Base36, Base58, Base62, Base64, Base64Url, Base85, Base91, Base92, and more with the best base encoding scheme decoding tool in town. It's useful for CTFs, Bug Bounty Hunting, and Cryptography. This tool will be able to accept single user input, multiple inputs from a file, input from argument, multi-encoded bases,bases in image EXIF data , bases on images with OCR and decode them incredibly fast.We are using 'anyase32' and similar libraries to work with Base encoded files. This application probably won't be for naive users as it involves cryptographic techniques which are almost never directly required to a regular person. It can be used by developers on their receiving end to be able to receive and decode any type of encoded information and use that information directly without hassle.

Keywords- *Base64, Base Decoding, Python, Cryptography*

I. INTRODUCTION

In the digitally advanced world we live in nowadays, Security and privacy are the most important aspects of life. Any data which is sent through any digital medium is being encoded in some form or the other. Applications that are highly data driven, Deal with different types of encoding on a daily basis. Hence they need to use different kinds of decoders for working with the data. We solved the problem by making a decoder that can decode any base encoded data with ease.

II. LITERATURE REVIEW

3.1 Dynamic decoders,
Authors: Michael A. Turi and José G. Delgado-Frias.
Dynamic decoders are always having advantages over static decoders because of their speed and power consumption. In this paper dynamic decoding schemes are discovered. Address decoders using selective pre-charging schemes are presented and analyzed here. These schemes are having advantage on simple decoder and the AND-NOR decoders. Results are also compared with conventional one and give satisfactory performance.

3.2 Nand gate architectures for memory decoder,
Authors: Shivkaran Jain, Arun kr. Chatterjee.
This paper has given some NAND gate design styles which when used in decoder reduces energy consumption and delay. Basically conventional, NOR style NAND, source coupled NAND is discussed. The three designs conventional, nor style NAND, source coupled NAND, ranges in area, speed and power. In nor style NAND transistors are added in parallel so high fan -in is obtained and logical effort is reduced. In source coupled NAND gate number of transistors is reduced it gives speed of operation compared to an inverter.

3.3 Universal Decoding Scheme,
Authors:Ireneusz B., Łukasz Z.
This paper universal decoding scheme is proposed. Universal decoders are made by an alternate stage of NAND and NOR gate which avoid unnecessary use of inverters. This paper overcomes the problem on simple decoders. AND gate are not available naturally, they are made up by using NOR and NOT gate.

3.4 Fast low-power decoders for RAMs,
Authors: B. S. Amrutur and M. A. Horowitz.
This paper explains low power SRAM techniques. Decoders with different logic styles are explained here. Modelling of the decoder is also explained here. Logical effort of circuits is calculated and according to that transistors are sized.

3.5 SRAM Basic Decoder,
Authors:Kevin Zhang.
This book/literature explains the basic structure of SRAM along with components. Basic design techniques of components are given and sizing issues are discussed. All SRAMs basic parts are covered along with their role in memory optimization.

III. PROBLEM STATEMENT

Applications that are highly data driven, Deal with different types of encoding on a daily basis. Hence they need to use different kinds of decoders for working with the data. We made a decoder that can work with any base-encoded data and decode it while being on the receiving end. It can also decode encoded files if provided. It can use OCR techniques to scan bases from images.

IV. PROPOSED SYSTEM

Alphanumeric Base Decoder is a tool written in Python that can decode all alphanumeric base encoding schemes. This tool can accept single user input, multiple inputs from a file, input from the argument, multi-encoded bases, bases in image EXIF data, bases on images with OCR and decode them incredibly fast. For a basic demo, try the Web Interface that uses Alphanumeric Base Decoder's API. Fun Fact! I initially made this after being fed up with lame CTF challenges with multi-encoded bases. Now some of them started doing that in Steganography challenges so I automated that too smh!

Features

- Decode multi-encoded bases of any pattern.
- Decode bases in image EXIF data.
- Decode bases on images with OCR detection.
- Can decode multiple base encodings from a file.
- Generate a wordlist/output with the decoded bases.
- Predicts the type of encoding scheme.

Supported Encoding Schemes

- Base16,
- Base32,
- Base36,
- Base58,
- Base62,
- Base64,
- Base64Url,
- Base85,
- Ascii85,
- Base91,
- Base92

V. SYSTEM STUDY

CRYPTOGRAPHY

8.2.1 Introduction to Cryptography:

The art of concealing information to induce secrecy in the communication and transmission of sensitive data is termed cryptography. Diving deep into the etymology of the word 'cryptography' shows that this word finds its origin in ancient Greek. Derived from words kryptos meaning "hidden" or "secret" and graphy meaning "writing", cryptography literally means writing something secretly.

The idea of cryptography is to convey a private message or piece of information from the sender party to the intended recipient without getting the message intruded on by a malicious or untrusted party. In the world of cryptography, this suspicious third party that is trying to sneak into a private communication to extract something sensitive out of it is called an adversary.

Cryptography protects us from these unwanted adversaries by offering a range of algorithms required to hide or protect our message in the best way possible and transmit it comfortably over a not-so-secure network.

Types of Cryptography:

We classify cryptographic practices into three types, considering the kinds of algorithms and keys used to secure the information.

Symmetric-Key Cryptography:

Symmetric-key cryptography has the same key for encrypting as well as decrypting the message. The sender is supposed to send the key to the recipient with the ciphertext. Both parties can communicate securely if and only if they know the key and nobody else has access to it.

Caesar cipher is a very popular example of symmetric key or secret key encryption. Some of the common symmetric key algorithms are DES, AES, and IDEA ETC.\

Symmetric-key systems are quite fast and safe. However, the drawback of this kind of communication is the protection of the key. Conveying the key secretly to all the intended recipients was a worrisome practice. Any third party knowing your key is a gruesome thought as your secret won't be a secret anymore. For this reason, Public-key cryptography was introduced.

Asymmetric-Key Cryptography:

Asymmetric-key or public-key cryptography involves two keys. One used for encryption called a public key and the other one used for decryption known as a private key. Now, only the intended recipient knows the private key.

The flow of this communication goes like this: Sender asks for your public key to encrypt his message with the help of it. He then forwards the encrypted message to the recipient.

The recipient receives the ciphertext, decodes it with the help of his private key, and accesses the hidden message.

AN ESSENCE TO ENCODING AND DECODING:

Encoding and decoding are used in many forms of communications, including computing, data communications, programming, digital electronics and human communications.

These two processes involve changing the format of content for optimal transmission or storage.
In computers, encoding is the process of putting a sequence of characters (letters, numbers, punctuation, and certain symbols) into a specialized format for efficient transmission or storage. Decoding is the opposite process -- the conversion of an encoded format back into the original sequence of characters.
These terms should not be confused with encryption and decryption, which focus on hiding and securing data. (We can encrypt data without changing the code or encode data without deliberately concealing the content). function) get multiplied multiple times as we move towards the starting layers. As a result of this, the gradient almost vanishes as we move towards the starting layers, and it becomes difficult to train these layers. Turns out that an RNN transforms the existing information completely by applying a function. Because of this, the entire information is modified, on the whole, i. e. there is no consideration for 'important' information and 'not so important' information.

ENCODING AND DECODING IN DATA COMMUNICATIONS:
Encoding and decoding processes for data communications have interesting origins. For example, Morse code emerged in 1838 when Samuel Morse created standardized sequences of two signal durations, called dots and dashes, for use with the telegraph. Today's amateur radio operators still use Q-signals, which evolved from codes the British Postmaster General created in the early 1900s to ease communication among British ships and coast stations.

Manchester encoding was developed for storing data on magnetic drums of the Manchester Mark 1 computer, built in 1949. In that encoding model, each binary digit, or bit, is encoded low then high, or high then low, for equal time. Also known as phase encoding, the Manchester process of encoding is used in consumer infrared protocols, radio frequency identification and near-field communication.

Encoding and Decoding Schemes:
There are various standard encoding schemes each part of data is assigned a unique code. Some of the popular encoding schemes are mentioned below:

Base16
Hexadecimal (or hex) is a base 16 system used to simplify how binary is represented. A hex digit can be any of the following 16 digits: 0 1 2 3 4 5 6 7 8 9 A B C D E F. Each hex digit reflects a 4-bit binary sequence.

Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Base32:
Base32 is the base-32 numeral system. It uses a set of 32 digits, each of which can be represented by 5 bits (25). One way to represent Base32 numbers in a human-readable way is by using a standard 32-character set, such as the twenty-two upper-case letters A–V and the digits 0-9. However, many other variations are used in different contexts.

Value	Char	Value	Char
0	A	16	Q
1	B	17	R
2	C	18	S
3	D	19	T
4	E	20	U
5	F	21	V
6	G	22	W
7	H	23	X
8	I	24	Y
9	J	25	Z
10	K	26	2
11	L	27	3
12	M	28	4
13	N	29	5
14	O	30	6
15	P	31	7

Base36:
Base36 is a binary-to-text encoding scheme that represents binary data in an ASCII string format by translating it into a radix-36 representation. The choice of 36 is convenient in that the digits can be represented using the Arabic numerals 0–9 and the Latin letters A–Z (the ISO basic Latin alphabet).

Binary Number	Decimal Number	Character B
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
...	11	B
...	12	C
...	13	D
...	14	E
...	15	F
	...	Z

Base64:

Base64 is an algorithm to convert a stream of bytes into a stream of printable characters (and back). The origin of such binary-to-text encoding schemes like Base64 is the requirement to send a stream of bytes over a communication channel which does not allow binary data but only text-based data.

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

VI. CONCLUSION

Through this project we develop a full solution to decoding data from a digital medium. Our application is designed to be on the receiving end of the transmission medium. Whenever an encoded file arrives through the medium, may it be of any format (img, txt, str), it can decode and extract the underlying contents. Developers normally need to use different decoders for different encoding schemes. Now onwards, developers can work with the data without getting into the hassle of different types of decoders.

Output Fig 1: Take an encoded string as an input & Decodes it.

Output Fig 2: It takes an image as an input and decodes and shows the message hidden in its Exif data..

VII. REFERENCES

[1] Michael A. Turi and José G. Delgado-Frias, "High-Performance Low-Power Selective Precharge Schemes for Address Decoders" IEEE Transactions On Circuits And Systems, vol. 62, no. 9, Page: 817 - 821, 2015.

[2] Shivkaran Jain, Arun kr. Chatterjee "Nand gate architectures for memory decoder" International Journal of Computers & Technology, Page: 610-614, 2015.

[3] I. Brzozowski, Ł. Zachara and A. Kos "Universal Design Method of n-to-2n Decoders," Mixed Design of Integrated Circuits and Systems Conference, Poland, June 2014.

[4] B. S. Amrutur and M. A. Horowitz, "Fast low-power decoders for RAMs," IEEE J. olid-State Circuits, vol. 36, no. 10, pp. 1506-1515, Oct. 2014.
[5] Bharadwaj S. Amrutur, "Design and Analysis Of Fast Low Power SRAMs" P.H.D Thesis, Stanford University, 2013.
[6] A. Pavlov, M. Sachdv "CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies": Springer publication. September 2014
[7] Kevin Zhang, "Embedded Memories for Nano-Scale VLSIs" Springer publication.
[8] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic, "Digital Integrated Circuits a Design Perspective," PHI Learning, 2009.
[9] V. Sharma "SRAM Design for Wireless Sensor Networks, Analog Circuits and Signal Processing" Springer Science, New York 2013.