Proposition and Implementation of PNS Algorithm With Data Compression

Harsh Khemka
School of Information and Technology, VIT University, Vellore, Tamil Nadu, India

Abstract
Cryptography is the secured technique for data communication in non-human readable text format between two parties in the presence of third party. The algorithm is different from other cryptographic algorithms since the main idea behind this algorithm is that any number can be viewed as the Pth number of N digit having the single digit sum as S. Therefore, the algorithm is said as PNS (Position-Number of Digit-Single sum). This algorithm can be used for encryption-decryption on any human-readable text therefore is a useful mechanism for data security via cryptography. I have effectively used data compression on it to provide better security. In the context where client-server secure communication exists, there must be 4 specific security requirements:

- **Authentication:** The process of ensuring one`s secure identity in the communication process.
- **Privacy:** The property of promising than the message cannot be interpreted by anyone except for the person it is intended for.
- **Integrity:** The property which ensures that the message received by the receiver is of supreme originality.
- **Non- repudiation:** A criteria to prove that the message is not a spam message, for that it is meaningful and sent by a valid sender with a valid reason.

Keywords
Authentication, Cryptography, Integrity, PNS Algorithm.

I. Introduction

A. Cryptography: A Brief Description

Cryptography has become the vital necessity for secured communication for sharing any valuable data or information where the information can be viewed by people for whom the information can prove to be non-integral, but the key for decrypting the encrypted information is only handled by the person for whom the information is served for.

B. Classification of Cryptography

The cryptographic algorithms are classified into three basic categories based on the number of keys used in the encryption and decryption process:

1. **Secret Key Cryptography:** The kind of cryptography which uses only one key both the encryption and decryption process.
2. **Public Key Cryptography:** The kind of cryptography which uses separate keys for encryption and decryption process.

C. Secret Key Cryptography (SKC):

Secret key cryptography is segregated into following streams:

1. **Stream Cipher:** Stream cipher particularly works on a single bit at a particular instant of time.
2. **Block Cipher:** Block cipher can work in few modes and few important are Electronic codebook(ECB), Cipher block chaining(CBC), Cipher feedback(CFB),Outputfeedback(OFB)

C. Public key cryptography

This type of cryptography is based on mathematicalproblems that have no efficient solution. It usually uses a public and private key for encryption and decryption this type of encryption is highly secure for transferring information through the internet.
Many classical algorithms like RSA, DES and AES are already available in the arena of cryptography. They all inherit the concept of converting the text into bytes which is a complicated process if you want in depth analysis of algorithm for implementation purpose.RSA algorithm is classifiedas Public Key Cryptography. RSA algorithm consumes a lot of time in the decryption process as it has to use multiple keys to decrypt the encrypted text. DES algorithm is an example of symmetric cryptography. In DES algorithm, for encryption of every 64bit text we have to use 56-bit decryption key, though the encrypted text is 64 bit but accomplishing this is quite a tedious job. Bitwise operation and modular approach over-romanticizes the situation rather than giving it a much more simplified approach furthermore increasing the complexity of the algorithm. Although both, DESand PNS follow Secret Key Cryptography, and arevulnerable to brute force attack but PNS is designed with logical approach rather than any bitwise operations, modular complexities which in turn can adhere much longer to the stimulus of brute force attack.The main problem which deterred the secure communication was lack of logical approach in the available algorithms.And another conflict was that the size of encryption and decryption keys was confined to particular range of bit size. These problems had feasible solutions which I have worked upon in the PNS algorithm. This idea can revolutionize the approach for designing cryptographic algorithms than pertaining only to mathematical and bitwise operation. Also in classical algorithms the encryption and decryption process is dependent on the architecture of the computer because of the restriction of block size as message is forwarded in particular blocks, so if we want to send a message which exceeds the block size then we need to break the message into parts and send it in two different consecutive blocks, while this is not the case in PNS algorithm as I have overcome the restricted message length or byte size. Hence PNS algorithm is favorable for connection oriented client- server architecture and also for electronic-mail communication

II. Fundamentals of PNS Algorithm

The fundamentals of the PNS algorithm can be derived from the name itself. `P` which signifies the position of the number having n digits which is computed from the `N` part of the algorithm having `S` as a single sum obtained from multiple summation of the digits. Hence the full form of the abbreviation is “Position of Number of Digits-Single sum”.
For instance:

- 1 is encrypted to 111.
- 2 is encrypted to 112
- 21 is encrypted to 323
- 126 is encrypted to 339

A. Stairway to Algorithm Encryption:

i. Take a number α.
ii. Calculate: α % 9
iii. Store the value of II in K.
iv. If K==9, then go to step V or else go to step VI.
v. S=0
vi. S= K
vii. Find the number of digit and store it in N.
viii. Calculate start = 10^{N-1}
ix. Repeat Step X to Step XI till start ≤ α
x. Calculate S in the similar way like it is calculated in Step II to VI.
xi. If S matches with the number increment P.
xii. Calculate: (P*100)+(N *10)+ S
xiii. Store the value of XII in β.

B. Decryption:

i. Take a number β.
ii. Calculate: β % 10.
iii. Store the value of II in S.
iv. Calculate:(β/10)%10.
v. Store the value of IV in N.
vi. Calculate:(β/100).
vii. Store the valueof VI in P.
viii. Calculate start = 10^{N-1}
ix. Take a value G=0.
x. Repeat Step X to Step XI till G≤P from start.
xi. If S==9 goto Step XII else goto step XIII.
xii. If Z ≥ start % 9==0 increment P.
xiii. If Z ≥ start % 9==S increment P.
xiv. Calculate the value of Z-1.
xv. Store the value of XIV asα.

α – Signifies original number.
β - Signifies encrypted number.
Z – Signifies integers.

III. Experimental Design

The PNS cryptographic algorithm can be implemented via mail server using java mail API by which we can also Email the message in encrypted form to any user, who on having the key for decryption can decrypt the encrypted text. This procedure can take place between multiple clients and server where the encrypted text will be passed to each and every client but it could only be decrypted by the intended client. The interface design consists of the text area for the message which is intended to be sent,
Highlighted advantages which PNS enjoys over other algorithms are:

- We don't have to convert the number in bytes for encryption process.
- The data compression technique simulated in this algorithm, makes it more secure and reliable.
- Less complicated, easy to understand and importing of heavy weight packages not required.

A. Working of Encryption-Decryption

For encryption I am finding the PNS form of the ASCII (American Code of Information Interchange) value of the each and every character passed over the network and then compressing the PNS form further to decrease the size of the cipher text. The data compression is done because it reducesthe cipher text from 3*L to XwhereL≤X≤2*L where L is the number of characters in the string. For decryption, first the data is decompressed and is converted to PNS form then 3 consecutive integers (PNS form of the ASCII value) starting from the first are read sequentially and are converted into their original ASCII values which thereafter are converted to meaningful text or message which was intended to deliver.

For example: Message: My room No is 124
Encrypted message: 8253343259212333233321325826233325136237325524525527
= *hK86Q+67-#/hT76I: _/JC/JG
Decryption:
*hK86Q+67-#/hT76I: _/JC/JG
8253343259212333233321325826233325136237325524525527
My room No is 124.

IV. Screenshots

```
harsh@hare-krishna:~/Desktop/ecryp$ java crypa
Enter your string
My email id is harshkhemka9@gmail.com
ORIGINAL TEXT =My email id is harshkhemka9@gmail.com
CIPHER TEXT   =*hK86I6-!1#S=6I:##/#T;6I9"^:-]9#g9#,5#g1T;+#@5"]:#u+"r7-A
harsh@hare-krishna:~/Desktop/ecryp$
```

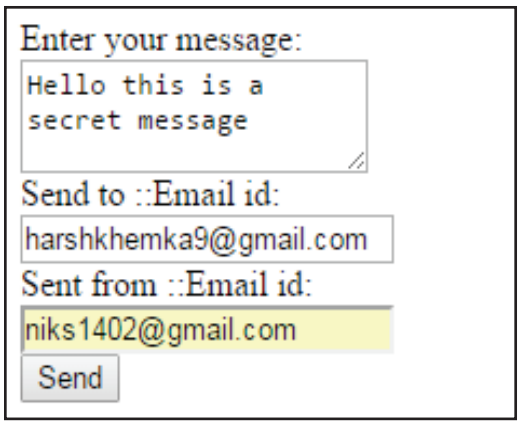
Fig. 1: Screenshot for Encryption

```
harsh@hare-krishna:~/Desktop/ecryp$ java crypd
Copy your cipher text
*hK86I6-!1#S=6I:##/#T;6I9"^:-]9#g9#,5#g1T;+#@5"]:#u+"r7-A
CIPHER TEXT=*hK86I6-!1#S=6I:##/#T;6I9"^:-]9#g9#,5#g1T;+#@5"]:#u+"r7-A
ORIGINAL TEXT=My email id is harshkhemka9@gmail.com
harsh@hare-krishna:~/Desktop/ecryp$
```

Fig. 2: Screenshot for Decryption

V. Experimental Analysis

To test the algorithm for it better value I have implemented in mail server using Java Mail API where I have to use javamail.jar and activation.jar to setup the mail environment in our system by the use of which I have converted our system into a mail server. I have mailed some message using JAVA mail API from one Gmail account to another Gmailaccount. In this process I have send the encrypted version of the text that has to be transferred. For doing this I have used JSP,Servlet, and Apache Tomcat Server. For sending mail, the sender email id and password both should be authenticated and verified.
Mail.jsp
The interface above can describe the very first page of our Mail Server. The mail is being transferred from *niks1402@gmail.com* to *harshkhemka9@gmail.com.*

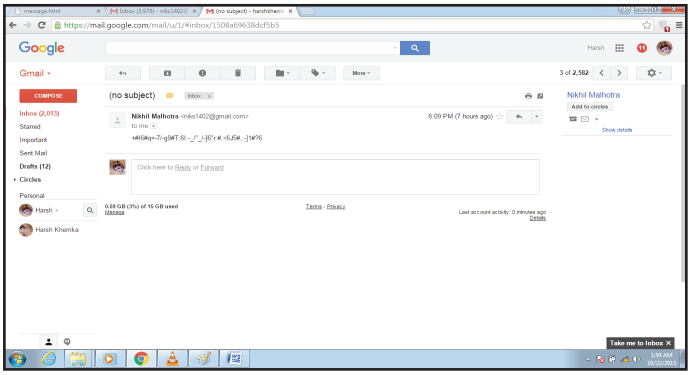


To confirm the user that the mail is transferred a page is displayed named as Success.jsp that the mail has been successfully sent.

Success.jsp

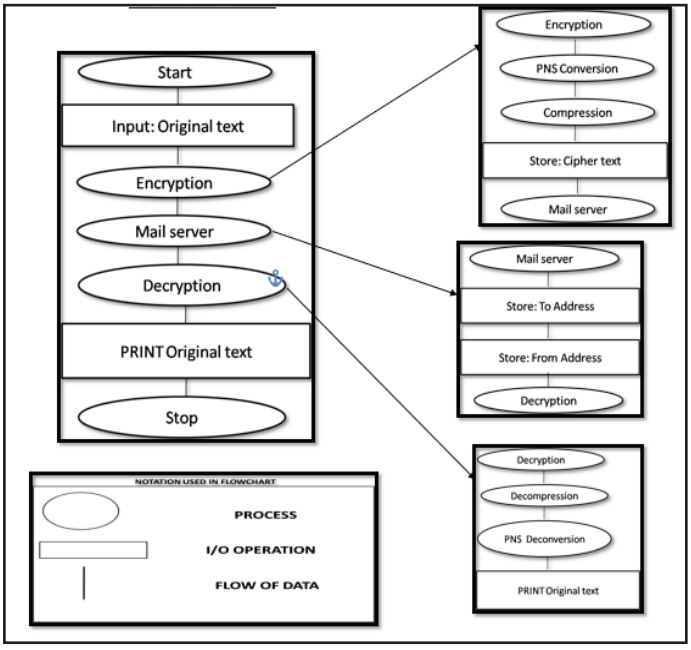
The mail is send in SendMail.java since it is the working part where I send the decrypted message therefore it has no interface.

Gmail account: *harshkhemka9@gmail.com*



This is the result of the experiment performed on the algorithm the user receive the message in encrypted form and now he can decrypt it using the decrypt module provided to him.

VI. Process Flowchart



VII. Performance Analysis

The performance of the PNS algorithm in comparison to different available algorithms is way above the standards for which any cryptographic algorithm needs to qualify as a supreme one. The PNS algorithm is relentlessly exceeding in efficiency, time constraint and flexibility thereby a withstanding force if experiencing minimal flaws. The algorithm encrypts the text in a logical approach which makes encryption faster and the encrypted text is compressed using data compression technique where two bit of the encrypted text is added with a unique number and converted into character.

The PNS algorithm is efficient when we talk about:

A. Speed

It is faster than the entire symmetric algorithms proposed till date such as DES, AES and others because it does not required complex bitwise operation and thereby encryption is done at a comparatively faster pace resulting in faster mode of execution.

B. Length of Text

The length of text is not limited to 128 bits or 64 bits as we have discussed above in AES and DES algorithms respectively. You can transfer any block of message using this algorithm. The algorithm has no length constraint or bit constraint.

3. Security

The algorithm is secured from implementation point of view because of the rigorous logic which can pose difficulty for rupture of code and above that the encrypted text is first compressed before being sending to client or someone.

4. Range

The range for any character is not limited; you can use any of the character which is readable by a human eye as your original message. Alphanumeric character, Abbreviated forms, Digits, Symbols, Punctuation marks and Operators all are differentiated using this algorithm.

5. Order

The algorithm sees to it that the order of the information has to be maintained i.e. each and every token of the message will be read after decryption in the similar manner as it was read before encryption.

6. No Data Loss

The algorithm sees to it that there is no data loss i.e. each and every token transferred in the mail should be received by the receiver.

VIII. Conclusion

I have done a lot R&D in the working of Java mail API and mail server to implement this algorithm for better, secure, and legit communication. Servlets and Java Server Pages are used for the interface and transfer of mail over the mail server. I have hope to pave a new path in the field of cryptography by decrypting the text without the decryption key. The algorithm defines its uniqueness by the way the number is seen in this algorithm which is the mere definition and the reason for which the name is PNS algorithm. The data compression technique where each two bit is combined and added with a unique integer has also put more weigh in the quality of the algorithm therefore, making it difficult for the third party to decipher it and hence fulfilling the objective of the cryptographic algorithm. Hence it can be considered as a new way of creating cipher text from a given text with least complexity and superlative, feather weight modules. The main focus of the algorithm is maintaining dual functionality which can help to forward certain amount of information with maximum security and by monitoring minimal time and memory space management for the encryption process and decryption process. The algorithm can be improvised further if the same PNS algorithm is made functional to loop N times, which can in turn enhance the integrity and security of data transmission. It would be an approach similar to AES algorithm. After the implementation of the algorithm N times, we can divide it by a large exponent of integer something

like and thereby generating an encrypted text. The encrypted text can then be compressed using simple compression techniques. The text generated via compression can be used in password parameter as encrypted password in the particular database of any kind of social networking, financial organization and many e-commerce websites. Hence, making a sound impact on the standard of efficiency which is expected from any cryptographic algorithm. Thereby having a supreme potential to fight its way into the market and find the spot nearby the classical symmetric algorithms such as DES and AES.

References

[1] Gary C. Kessler,"An Overview of Cryptography", 8th July 2015
[2] Ferguson, N., B. Schneier,"Proclicol Cryptography", Wiley & Sons, 2003
[3] ArunKejariwal,“Crypticprimes”,Potentials, 2004.
[4] X. Lai, J. L. Massey,“A proposal for a new block encryption standard”, In Advances in Cryptology-EUROCRYPT, 1990.
[5] Najib A. Kofahil, Turki Al-Somani, Khalid Ai-Zamil, "Performance Evaluation of Threeencryption/Decryption Algoriithms".
[6] Wayne G. Barker,“Introduction to the analysis of the Data Encryption Standard (DES)”, A cryptographic series, 1991.
[7] M. J. B. Robshaw,“Block Ciphers”, Technical Report, RSA Laboratories, Number TR - 601, 1994
[8] M. J. B. Robshaw,“Stream Ciphers”, Technical Report, RSA Data Security, Inc., Number TR-701, July 1995.
[9] D K-AHN,“The Codebreakers”, MacmillanPublishing Company, New York, 1967.
[10] Michael C.-J. Lin, Youn-Lonlg Lin, “A VLSI Implementation of the DES Encryption/Decryption Algorithm”.
[11] Jose Luis Chao,“DES implementation in Texas Instruments MSP430 Microcontroller”.

Harsh Khemka has done Bachelor Of Computer Application from Meghnad Saha Institute Of Technology and is currently pursuing Masters of Computer Application in VIT University, Vellore.
He has a keen interest in the world of security where we have to deal with various techniques to secure ourselves.
Harsh is still researching on new kind of algorithm which can help us to secure in the world of vast growing internet and hence provide security to all of us and for us.