

Non Invertive Encryption in Reentrant Layered Logic Tables (RLLT)

Colin James III, PhD, Principal Scientist, email: info@cec-services.com
CEC Services, LLC, 1613 Morning Dr, Loveland CO 80538-4410

Keywords

Logic Table Technology, LTT, Layered Logic Tables, LLT, non invertive encryption, NIE, Reentrant Layered Logic Tables, RLTT, real time performance, RTP, SQL

Abstract

Reentrant layered logic tables (RLLT) form the basis of an encryption engine implemented in structured query language (SQL). The theoretical method chosen is non invertive encryption (NIE) known to be the strongest cryptography. The text key for encryption is in blocks of 128 characters and is derived from public text. On a desktop computer, the processing time in SQL for 128 levels of logic is about 0.70 seconds. The text key output is shown to be random.

Non Invertive Encryption (NIE)

Non invertive encryption uses a secret method and keys that may not be reversed without the original method and keys. The initial encryption key is contained in two parts: the number of the text key; and the index number of the entry point into the text key. The text key is any set of text owned by the sender and receiver such as the Manhattan telephone directory or T*he King James Translation of the Holy Bible*. The text key may be a public, as opposed to a private, document. If the text key is public, then the index point is private. Conversely, if the text key is private, then the index key may be public.

The indexed point of entry into the text key is a private key number of 0 through 127 consistent with the 128 characters of the 7-bit ASCII set. The text key serves as a unique logic table that is self-indexing, after the reentrant layered logic table (RLLT) of James 2003.5.

The output numbers index an entire row of encryption text. To encrypt plain text of 16384 characters, with a table of 128 rows the number of possible row combinations is 128! or about 2^716. The square root of that number as about 2^358 is the practical number of secure combinations which alone is probably large enough to thwart a brute force attack. A more accurate number of secure combinations should take into account the secret key indexing the text key as the square root of (128!)^128 or about 2^45834.

Implementing NIE

The successive levels of output are such that level 1 is performed 128 times, level 2 is performed 127 times, and level 128 only 1 time. To avoid repeating select statements for successive levels, the output results for previous levels may be saved and reused as input

to subsequent levels. A way to avoid discrete queries for each level is to use only one logic query for all of the 128 levels as below.

```
CREATE TABLE key ( num INTEGER NOT NULL) @
CREATE TABLE perf ( dt TIMESTAMP NOT NULL) @
CREATE TABLE out ( string VARCHAR( 254), row_id INTEGER) @
CREATE TABLE logic ( index INTEGER NOT NULL, sub01 VACHAR( 2048)) @

INSERT INTO logic ( index. sub01) VALUE ( 1, '1xxx1xxxxxx1x11x111xx…') @
…
INSERT INTO key
  ( row_id, string )
VALUES
  ( ( 1+ SELECT COUNT( *) FROM key) / 2,
  ( SELECT S1.sub01 FROM logic AS S1
  WHERE S1.index IN
    ( SELECT T255.index FROM logic AS T255, logic AS T254
    WHERE   SUBSTR( T255.sub01, T254.index, 1)  <>  '' AND T254.index IN...
    ( SELECT T253.index FROM logic AS T253, logic AS T153 …
    ( SELECT T01.index FROM logic AS T01
    WHERE SUBSTR( T01.sub01, [key_text_num], 1)  <>  '')) … ) @
```

Performance in Real Time

The 128 encryption text keys for 128 levels of logic are processed in about 0.70 seconds on a 733 MHz box desktop computer with 768 MB RAM and 80 MB hard disk.

Measure of Randomness

Below is the source code to mirror the SQL logic for 128 levels. The output is plotted in the RLLT graph for comparison with a Random graph that follow.

```
!  SQL Graph - TrueBASIC® Source Code for graph
!
!  Visual measure of randomness by graphing output
!
!  © Copyright 2003 by CEC Services, LLC  All Rights Reserved

LIBRARY "HEXLIB.TRC"
DECLARE DEF Xor

DIM plaintxt$ ( 0:127, 0:127), plaintxt ( 0:127, 0:127)  ! text to encrypt
DIM ciphertxt$( 0:127, 0:127), ciphertxt( 0:127, 0:127)  ! encrypted text
DIM keytxt$   ( 0:127, 0:127), keytxt   ( 0:127, 0:127), keyid( 0:127, 0:127)
DIM diphertxt$( 0:127, 0:127), diphertxt( 0:127, 0:127)  ! decrypted text
DIM                            differs  ( 0:127, 0:127)  ! decryption test

DO WHILE MORE DATA
   READ txt0$
   LET txt1$ = txt1$ & txt0$
LOOP
LET txt1$ = txt1$ & txt1$ & txt1$ & txt1$
```

```
LET counter = 0
FOR i = 0 to 127
    FOR j = 0 to 127
        LET counter = counter + 1
        LET plaintxt$( i, j) =      txt1$[ counter: counter]
        LET plaintxt ( i, j) = ord( txt1$[ counter: counter])
    NEXT j
NEXT i

LET counter = 0
FOR i = 0 to 127
    FOR j = 0 to 127
        LET counter = counter + 1
        LET keytxt$( i, j) = txt1$[ counter: counter]
    NEXT j
NEXT i

FOR i = 0 to 127
    FOR j = 0 to 127
        LET keytxt( i, j) = ord( keytxt$( i, j))
    NEXT j
NEXT i

FOR h = 0 to 127
    FOR i = 0 to 127
        FOR j = 0 to 127
            LET keyid( i, j) = keytxt( i, keytxt( i, j))
        NEXT j
    NEXT i
NEXT h

FOR i = 0 to 127
    FOR j = 0 to 127
        ! Output character refers to character key
        !   LET ciphertxt ( i, j) = Xor ( plaintxt( i, j) , keyid( i, j))
        !   LET ciphertxt$( i, j) = CHR$( ciphertxt( i, j))
        ! Output character refers to row key
            LET ciphertxt ( i, j) =
                Xor ( plaintxt( i, j) , plaintxt( keyid( i, j), j))
            LET diphertxt ( i, j) =
                Xor ( ciphertxt( i, j), plaintxt( keyid( i, j), j))
        LET ciphertxt$( i, j) = CHR$( ciphertxt( i, j))
        LET diphertxt ( i, j) = Xor ( ciphertxt( i, j), keyid( i, j))
        LET diphertxt$( i, j) = CHR$( diphertxt( i, j))
    NEXT j
NEXT i

!MAT differs = diphertxt – plaintxt ! Decryption test
!IF difference <> 0 then
!   PRINT "Decryption failed"
!ELSE
!   PRINT "Decryption passed"
!END IF
!STOP

SET WINDOW 1, 128, 1, 128
PRINT "Random graph follows - press any key; for next graph press key again"
GET KEY k
CLEAR                           ! Screen
FOR i = 0 to 127
    FOR j = 0 to 127
        PLOT POINTS : rnd * ( i + 1), rnd * ( j + 1)  ! Random
```

```
    NEXT j
NEXT i
GET KEY k
CLEAR                               ! Screen
PRINT "Key graph follows - press any key; for next graph press key again"
GET KEY k
CLEAR                               ! Screen
FOR i = 0 to 127
    FOR j = 0 to 127
        PLOT POINTS : rnd * ( i + 1), rnd * ( keytxt( i, j) + 1)     ! Key
    NEXT j
NEXT i
GET KEY k
CLEAR                               ! Screen
PRINT "Cipher graph follows - press any key; for next graph press key again"
GET KEY k
CLEAR                               ! Screen
FOR i = 0 to 127
    FOR j = 0 to 127
        PLOT POINTS : rnd * ( i + 1), rnd * ( ciphertxt( i, j) + 1)  ! Cipher
    NEXT j
NEXT i

DATA "1: In the beginning was the Word, and the Word was with God, and the
Word was God."
DATA "2: The same was in the beginning with God."
DATA "3: All things were made by him; and without him was not any thing made
that was made."
DATA "4: In him was life; and the life was the light of men."
DATA "5: And the light shineth in darkness; and the darkness comprehended it
not."
DATA "6: There was a man sent from God, whose name was John."
DATA "7: The same came for a witness, to bear witness of the Light, that all
men through him might believe."
DATA "8: He was not that Light, but was sent to bear witness of that Light."
DATA "9: That was the true Light, which lighteth every man that cometh into
the world."
DATA "10: He was in the world, and the world was made by him, and the world
knew him not."
DATA "11: He came unto his own, and his own received him not."
DATA "12: But as many as received him, to them gave he power to become the
sons of God, even to them that believe on his name:"
DATA "13: Which were born, not of blood, nor of the will of the flesh, nor of
the will of man, but of God."
DATA "14: And the Word was made flesh, and dwelt among us, (and we beheld his
glory, the glory as of the only begotten of the Father,) full of grace and
truth."
DATA "15: John bare witness of him, and cried, saying, This was he of whom I
spake, He that cometh after me is preferred before me: for he was before me."
DATA "16: And of his fulness have all we received, and grace for grace."
DATA "17: For the law was given by Moses, but grace and truth came by Jesus
Christ."
DATA "18: No man hath seen God at any time; the only begotten Son, which is
in the bosom of the Father, he hath declared him."
DATA "19: And this is the record of John, when the Jews sent priests and
Levites from Jerusalem to ask him, Who art thou?"
DATA "20: And he confessed, and denied not; but confessed, I am not the
Christ."
DATA "21: And they asked him, What then? Art thou Elias? And he saith, I am
not. Art thou that prophet? And he answered, No."
DATA "22: Then said they unto him, Who art thou? that we may give an answer
to them that sent us. What sayest thou of thyself?"
```

```
DATA "23: He said, I am the voice of one crying in the wilderness, Make
straight the way of the Lord, as said the prophet Esaias."
DATA "24: And they which were sent were of the Pharisees."
DATA "25: And they asked him, and said unto him, Why baptizest thou then, if
thou be not that Christ, nor Elias, neither that prophet?"
DATA "26: John answered them, saying, I baptize with water: but there
standeth one among you, whom ye know not;"
DATA "27: He it is, who coming after me is preferred before me, whose shoe's
latchet I am not worthy to unloose."
DATA "28: These things were done in Bethabara beyond Jordan, where John was
baptizing."
DATA "29: The next day John seeth Jesus coming unto him, and saith, Behold
the Lamb of God, which taketh away the sin of the world."
DATA "30: This is he of whom I said, After me cometh a man which is preferred
before me: for he was before me."
DATA "31: And I knew him not: but that he should be made manifest to Israel,
therefore am I come baptizing with water."
DATA "32: And John bare record, saying, I saw the Spirit descending from
heaven like a dove, and it abode upon him."
DATA "33: And I knew him not: but he that sent me to baptize with water, the
same said unto me, Upon whom thou shalt see the Spirit descending, and
remaining on him, the same is he which baptizeth with the Holy Ghost."
DATA "34: And I saw, and bare record that this is the Son of God."
DATA "35: Again the next day after John stood, and two of his disciples;"
DATA "36: And looking upon Jesus as he walked, he saith, Behold the Lamb of
God!"
DATA "37: And the two disciples heard him speak, and they followed Jesus."
DATA "38: Then Jesus turned, and saw them following, and saith unto them,
What seek ye? They said unto him, Rabbi, (which is to say, being interpreted,
Master,) where dwellest thou?"
DATA "39: He saith unto them, Come and see. They came and saw where he dwelt,
and abode with him that day: for it was about the tenth hour."
DATA "40: One of the two which heard John speak, and followed him, was
Andrew, Simon Peter's brother."
DATA "41: He first findeth his own brother Simon, and saith unto him, We have
found the Messias, which is, being interpreted, the Christ."
DATA "42: And he brought him to Jesus. And when Jesus beheld him, he said,
Thou art Simon the son of Jona: thou shalt be called Cephas, which is by
interpretation, A stone."
DATA "43: The day following Jesus would go forth into Galilee, and findeth
Philip, and saith unto him, Follow me."
DATA "44: Now Philip was of Bethsaida, the city of Andrew and Peter."
DATA "45: Philip findeth Nathanael, and saith unto him, We have found him, of
whom Moses in the law, and the prophets, did write, Jesus of Nazareth, the
son of Joseph."
DATA "46: And Nathanael said unto him, Can there any good thing come out of
Nazareth? Philip saith unto him, Come and see."
DATA "47: Jesus saw Nathanael coming to him, and saith of him, Behold an
Israelite indeed, in whom is no guile!"
DATA "48: Nathanael saith unto him, Whence knowest thou me? Jesus answered
and said unto him, Before that Philip called thee, when thou wast under the
fig tree, I saw thee."
DATA "49: Nathanael answered and saith unto him, Rabbi, thou art the Son of
God; thou art the King of Israel."
DATA "50: Jesus answered and said unto him, Because I said unto thee, I saw
thee under the fig tree, believest thou? thou shalt see greater things than
these."
DATA "51: And he saith unto him, Verily, verily, I say unto you, Hereafter ye
shall see heaven open, and the angels of God ascending and descending upon
the Son of man."

END
```

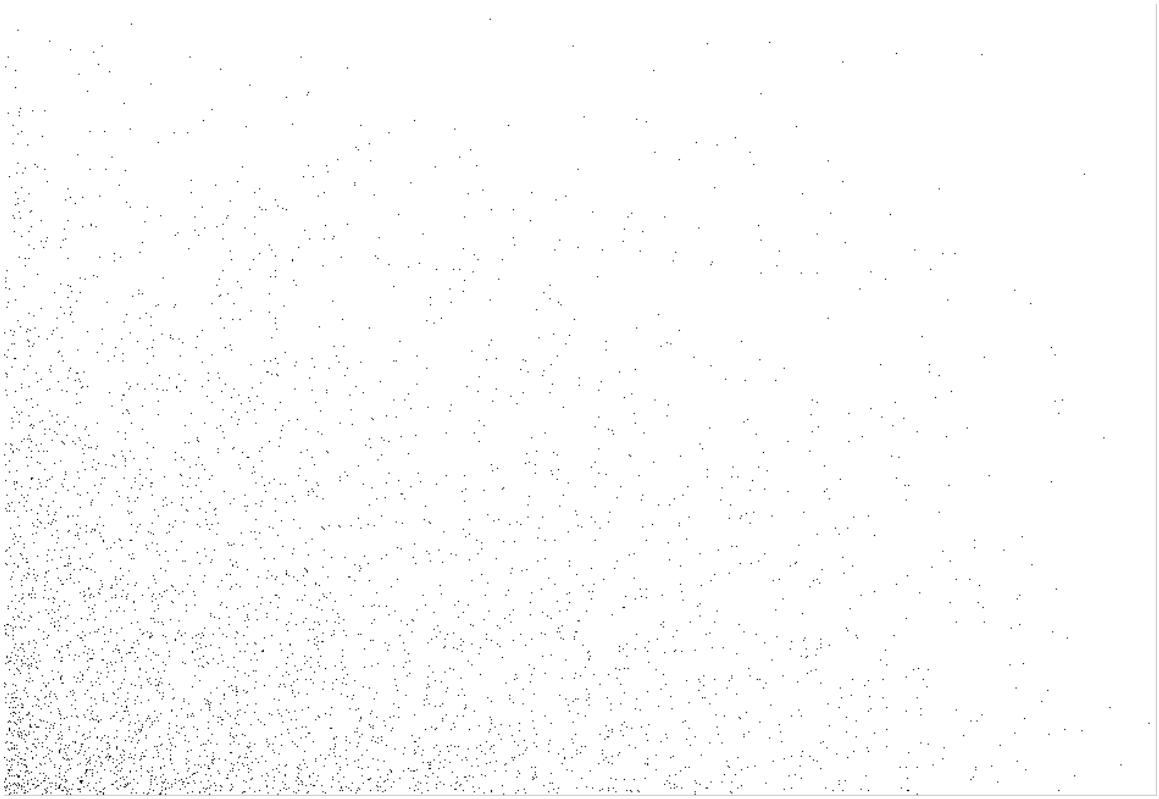
The text from which the text keys are derived is John 1:1-51, put into data blocks for each verse. The RND function of TrueBASIC® in the computer program is known mathematically to generate very good pseudo random numbers. The random distribution may be shown by turning on pixels for scatter such as the Random Graph below. The RND function uses the same seed to produce the same sequence of random numbers in the graphs below.

In the Random Graph and the Key Graph, there are no obvious visual patterns of streaking or clumping. The dot distribution for the Key Graph is essentially that of the Random Graph. The points in the Random Graph appear less dense than the Key Graph because the number of random points is not constrained by the frequency distribution of 128 potential characters of the text keys, some of which are not present in the text from which the text keys are derived.

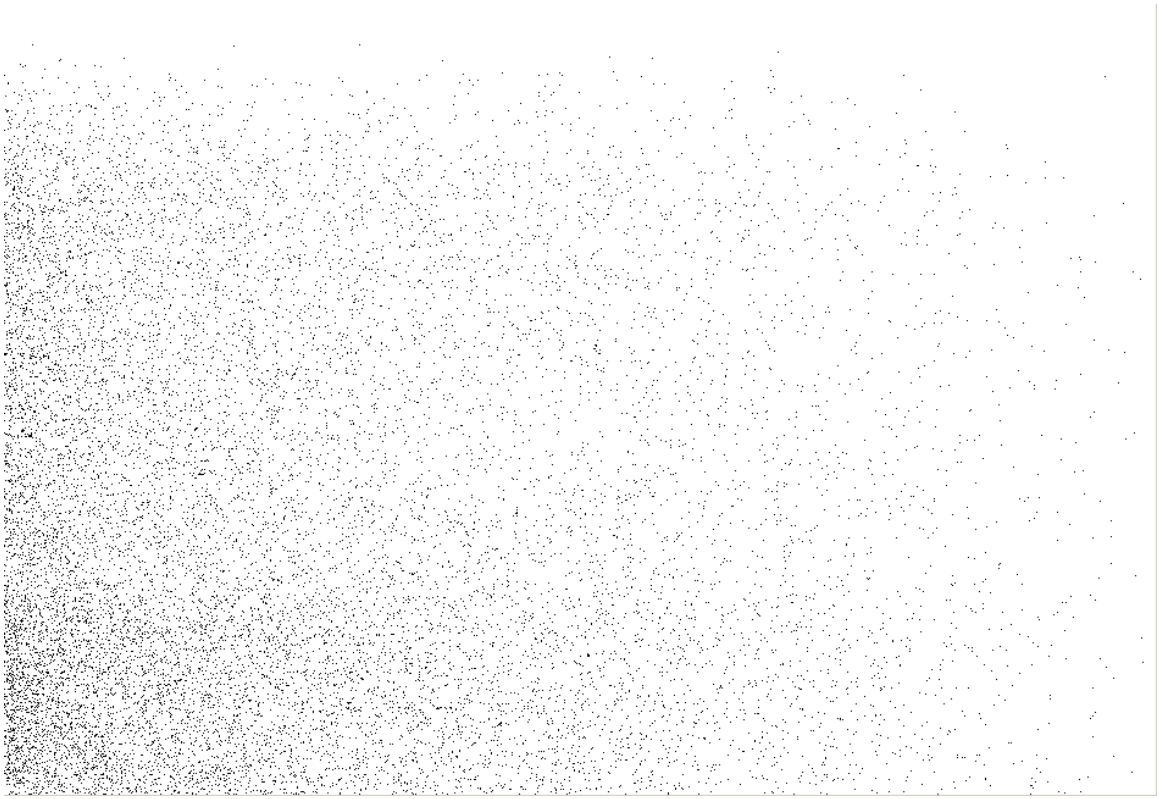
The relative height of the graphs also differs. The Random Graph is slightly higher because in the y-axis the random numbers are 1 through 128. The Key Graph is slightly shorter because in the y-axis of the distribution there are the 128 characters of text keys, some of which are not present in the text from which the text keys are derived.

The advantage of using a non random source of text key is that it is ubiquitous and hence convenient. The disadvantage is that characteristics about the source of the text keys ultimately may be discovered such as higher frequencies of certain characters for in this case Elizabethan English. However, using a text key only once per encryption session minimizes that attack.

In this analysis of randomness, the plain text to be encrypted is the same as that basis for the text keys. However any plain text could be used. Therefore a graph of the encrypted plain text or cipher text to show randomness, as available in the computer program, is not of particular interest and not reproduced here.



Random Graph



Key Graph

Conclusion

Encryption with RLLT has the advantages that it is:

1. Defined as non invertive, meaning it is not reversible as in public key;
2. Based on public text key and private index key for ease and security;
3. Minimized for computational intensity, unlike public key;
4. Designed with one reentrant logic table of 64 columns and 64 rows;
5. Implemented in one line of complex SQL code and thus 100% portable;
6. Proved statistically to output random text keys; and
7. Required to perform in real time.

Vertical market implementations are to be isolated and discussed in forthcoming papers.

Acknowledgments

Thanks are due to Larry Cagg of Cagg Enterprises for helpful discussions.

References

James, C. 2003.5, “Implementing Performance in Reentrant Layered Logic Tables (RLLT)”, unpublished, CEC Services, LLC, Loveland CO.

James, C. 2003.4, “Reentrant Layered Logic Tables (RLLT)”, [in submission to the Industrial Water Conference 2003 (IWC2003)], CEC Services, LLC, Loveland CO.

James, C. 2003.3, “Layered Logic Tables (LLT)”, unpublished, CEC Services, LLC, Loveland CO.

James, C. 2003.2, "Software Factory", unpublished brochure, CEC Services, LLC, Loveland CO.

James, C. 2003.1, "Software Factory", unpublished poster, CEC Services, LLC, Loveland CO.

James, C. 2002.5, "The Software Development Methodology [SDM]", unpublished, CEC Services, LLC, Loveland CO.

James, C. 2002.4, "Reentrant Logic Table Technology", unpublished, CEC Services, LLC, Loveland CO.

James, C. 2002.3, "Static and Dynamic Driver Triggers", unpublished, CEC Services, LLC, Loveland CO.

James, C. 2002.2, "Additional Information", unpublished, CEC Services, LLC, Loveland CO.

James, C. 2002.1, “Implementation Details for Multiple Billing”, CEC Services, LLC, Loveland CO.

James, C. 2001.2, "Report Accounts [RA] v 1.2 Inventory / Point of Sale", unpublished, CEC Services, LLC, Loveland CO.

James, C., 2001.1, "Report Accounts [RA] v 1.2", unpublished, CEC Services, LLC, Loveland CO.

James, C. 1999.1, “Recent Advances in Logic Tables for Reusable Database Engines”, Proceedings of the American Society of Mechanical Engineers International, Petroleum Division, 75th Anniversary Conference, Energy Sources Technology Conference & Exhibition, Houston, Texas.

James, C., 1998.5, "Multiple and Self-Modifying Logic Tables with Queries", unpublished, CEC Services, LLC, Loveland CO.

James, C. 1998.4, “A Reusable Database Engine for Accounting Arithmetic”, Proceedings of The Third Biennial World Conference on Integrated Design & Process Technology, Vol. 2, pp. 25-30, Berlin, Germany.

James, C., 1998.3, "Competency test for CEC Services, LLC", unpublished, CEC Services, LLC, Loveland CO.

James, C. 1998.2, “Theory and Application of Logic Tables in Relational Database Engines”, Doctoral Dissertation, Pacific Western University, Los Angeles.

James, C., 1998.1, "Ticket Reservations [TR] ver 1.1", unpublished, CEC Services, LLC, Loveland CO.

James, C., 1997.2, "User Documentation", unpublished, CEC Services, LLC, Loveland CO.

James, C., 1997.1, "Logic Table Design for Reports in RA", unpublished, CEC Services, LLC, Loveland CO.