Adoption and Foster Care
AFCARS
Analysis and Reporting System

Technical Bulletin #7 Encryption Guidelines and Algorithms REVISED: January, 2012[1, 2]

This Technical Bulletin supplements information in Appendices A and B to 45 CFR 1355 on assigning record numbers to detailed foster care and adoption data submitted to the Children's Bureau. This Bulletin provides title IV-E agencies with guidance in encrypting record numbers in the Adoption and Foster Care Analysis and Reporting System (AFCARS) data file submissions.

Table of Contents

[1] Initial issuance date June 30, 1994, as Technical Bulletin #6, Encryption versus Sequencing; and March 9, 1995, as Technical Bulletin #9, Encryption Guidelines & Algorithms.
[2] Previously issued as Technical Bulletin #4.

I. Introduction

A. AFCARS Background

AFCARS is designed to collect uniform, reliable information on children who are under the responsibility of the title IV-B/IV-E agency for placement, care, or supervision. The collection of adoption and foster care data is mandated by section 479 of the Social Security Act (the Act). The requirements for AFCARS are codified in Federal regulation at 45 CFR 1355.40. Effective October 1, 2009, section 479B(b) of the Act authorizes direct Federal funding of Indian Tribes, Tribal organizations, and Tribal consortia that choose to operate a foster care, adoption assistance and, at Tribal option, a kinship guardianship assistance program under title IV-E of the Act. On January 6, 2012, the Administration for Children and Families (ACF) issued an Interim Final Rule (IFR)[2] to implement statutory provisions related to the Tribal title IV-E program. The Federal regulations at 45 CFR 1355.40 were amended to apply the same regulatory requirements for data collection and reporting to a Tribal title IV-E agency as are applied to a State title IV-E agency.

AFCARS was established to provide data that would assist in policy development and program management. Data can be used by policymakers at the Federal, Tribal, and State levels to assess the reasons why children are in foster care and to develop strategies to prevent their unnecessary placement into foster care. Specifically, the data include information about foster care placements, adoptive parents, and length of time in foster care, and make it possible to identify trends in particular geographic areas. Also, the data enable the Children's Bureau to administer the Federal title IV-E foster care and adoption assistance programs more effectively. The Children's Bureau and ACF use these data for a number of purposes, including:

- responding to Congressional requests for current data on children in foster care or those who have been adopted;
- responding to questions and requests from other Federal departments and agencies, including the General Accounting Office (GAO), the Office of Management and Budget (OMB), the Department of Health and Human Services' Office of Inspector General (OIG), national advocacy organizations, States, Tribes, and other interested organizations;
- developing short and long-term budget projections;
- developing trend analyses and short and long-term planning;
- targeting areas for greater or potential technical assistance efforts, for discretionary service grants, research and evaluation, and regulatory change; and
- determining and assessing outcomes for children and families.

Additionally, the AFCARS data are used specifically in the

- Adoption Incentives Program;
- Child Welfare Outcomes Report;
- Child and Family Services Reviews (CFSRs);

[2] 77 FR 896 (January 6, 2012)

- Title IV- E Eligibility Reviews; and
- Allotment of funds in the Chafee Foster Care Independence Program (CFCIP).

B. Overview of the Technical Bulletin

To protect confidentiality when reporting detailed data to AFCARS, title IV-E agencies are required to assign a sequential number or a unique number which follows the child as long as he or she is in foster care and use a sequential or unique record number for reporting adoptions.[3] While title IV-E agencies may use either sequential numbering or a unique number, the Children's Bureau encourages title IV-E agencies to use encryption of the agency's person number. This Technical Bulletin provides information on encryption methodologies that the agency can use and additional references on encryption. An agency may either develop its own encryption routine or use the encryption/decryption utility provided by the Children's Bureau. Information on the use of the utility can be found in Technical Bulletin #8.

II. General Information on Record Numbers

A. Background

The AFCARS data file includes an element for the record number (foster care element #4 and adoption element #3), and it must be a twelve digit number that cannot be linked to the child except at the title IV-E agency or local level. The record number is defined in regulations as:

<u>Foster care</u>

The sequential number which the title IV–E agency uses to transmit data to the Department of Health and Human Services (DHHS) or a unique number which follows the child as long as he or she is in foster care. The record number cannot be linked to the child's case I.D. number except at the title IV–E agency level.

<u>Adoption</u>

The sequential number which the title IV-E agency uses to transmit data to DHHS. The record number cannot be linked to the child except at the title IV-E agency level.

Title IV-E agencies may use a sequential number but, as noted above, the Children's Bureau encourages agencies to encrypt the person identification number. This approach is supported by the requirements for an encrypted, unique person identification number record number in the National Youth in Transition Database (NYTD). For purposes of the Federal child and family services reviews and the title IV-E foster care eligibility reviews[4], title IV-E agencies must be able to identify the case file for a specific record for up to three years after the end of the report period. **The title IV-E agency cannot use the child's actual person number or the child's social security number as the AFCARS record number.**

[3] See 45 CFR 1355, Appendix A Section II,I.D. and Appendix B, Section II.I.C.
[4] See 45 CFR 1355.31-37, and 1356.71.

Encryption makes it easier for title IV-E agencies to cross reference files for the identification of case record numbers at a later date. Also, title IV-E agencies are encouraged to use the same person record number for a child each time the child enters and exits foster care or receives services. This, combined with the use of the same ―key‖ allows the title IV-E agency and the Children's Bureau to develop cohort information and an annual database by matching records submitted for each of the report periods in a fiscal year. This is especially important in regard to the child and family services review and the statewide assessment. The statewide assessment includes AFCARS data from the three most recent and available Federal Fiscal Years.

B. Encryption Guidelines

Title IV-E agencies may either use the methods outlined in this document to encrypt record numbers or develop their own algorithms. However, all methods must adhere to the following guidelines:

Guidelines
• Each AFCARS record number must be unique. • The method selected must give the same result every time the procedure is executed. In other words, a client ID must always yield the same encrypted AFCARS record number after processing. • To enhance security, the encryption method should apply a minimum of two different algorithms to the record numbers. • Federal auditing standards require that the title IV-E agency be able to reverse the process and determine the original client ID for any AFCARS record number. The title IV-E agency must maintain any algorithms, tables, and keys used to generate the unique ID. Avoid functions and algorithms that, when reversed, can return more than one value. For example: ➢ functions such as $x=y^2$ should be used with care as the inverse (the Ö) can yield two results. For example, √4 = 2 or -2. (In this the user may want to specify |√|.) ➢ avoid algorithms that involve rounding since the result of two different operations may be rounded to the same number. For example: 8888/47 = 189.106 :rounded to 189 8889/47 = 189.127 :rounded to 189 • The method of encryption must be kept secure by the title IV-E agency. Only authorized personnel should have access to it. • The AFCARS record number, as cited in Technical Bulletin #9, AFCARS File Format, must be right justified and padded with zeros to fill out the 12 character field.

- Every character in the title IV-E agency's client ID should be transformed.
- Any of the characters in the ASCII character set may be used except:
    - control characters (ASCII codes 0 through 31). Control characters are excluded to prevent errors during Children's Bureau processing and parsing of the file.
    - ASCII characters 35 (#), 36 ($), 37 (%), and 64 (@). These characters are reserved as file identifiers, as indicated in the revised *Technical Bulletin #2, AFCARS File Format*.
- Each encrypted number must be scanned to ensure that the ASCII control characters (0-31) and ASCII characters 35, 36, 37, and 64 are not being used.

AFCARS record numbers may be encrypted using a variety of methods. In the following three sections, we will introduce algorithms title IV-E agencies may select from to encrypt record numbers:

- Substitution
- Transposition
- Bit-level operations

III. Substitution Method

A. Introduction

In this method of encryption, each digit in the ID is replaced by another digit generated through an algorithm. To see how this type of encryption is accomplished, consider the following number line:

```
  0 1 2 3 4 5 6 7 8 9
L <-------------------> R
```

A very simple encryption is accomplished by substituting each digit with the digit found 4 positions to the left on the number line. If the count of 4 extends beyond the number line, circle around and continue counting. By applying this rule, 1234 becomes 7890. *Please note that this example is for illustration purposes only. It is not sufficiently complex for encrypting the title IV-E agency client ID.*

The following substitution models (sections II.B.-II.F.) are appropriate for encrypting the title IV-E agency client IDs. However, if one of these substitution methods is chosen, the title IV-E agency should develop its own rules for substituting one digit for another. Please do not use the examples given as the substitution scheme for title IV-E agency encryption rules.

B. Unique Digit Substitution

Apply a different substitution rule for each digit. For example, 1234 is encrypted as 6493 using the number line and the following table of rules:

0 1 2 3 4 5 6 7 8 9
L <————————————> R

1st digit left:	5 positions
2nd digit right:	2 positions
3rd digit right:	6 positions
4th digit left:	1 position

On the number line, locate 1 and move five positions to the left to get 6.
On the number line, locate 2 and move two positions to the right to get 4.
On the number line, locate 3 and move six positions to the right to get 9.
On the number line, locate 4 and move one position to the left to get 3.

C. Variable Length Substitution

If the client ID length is variable, different rules can be applied based on the number of digits in the ID. For example, if the title IV-E agency's client ID can vary from 6 to 9 digits, the following table of rules could be applied:

Number of digits in State/Tribe ID	Shifting Pattern
6	1st digit right: 3 positions 2nd digit right: 5 positions 3rd digit left: 3 positions 4th digit right: 7 positions 5th digit left: 1 position 6th digit left: 4 positions
7	1st digit left: 6 positions 2nd digit right: 4 positions 3rd digit left: 2 positions 4th digit left: 8 positions 5th digit left: 3 positions 6th digit left: 9 positions 7th digit left: 2 positions
8	1st digit right: 1 position 2nd digit left: 3 positions 3rd digit left: 2 positions 4th digit right: 4 positions 5th digit left: 8 positions 6th digit right: 9 positions 7th digit left: 3 positions 8th digit right: 9 positions
9	1st digit left: 7 positions 2nd digit left: 4 positions 3rd digit left: 2 positions 4th digit right: 9 positions 5th digit left: 9 positions 6th digit left: 2 positions 7th digit right: 4 positions 8th digit left: 5 positions 9th digit right: 9 positions

D. Substitution with Shuffled Digits

Use a number line with shuffled digits. For example, 1234 is encrypted as 6361 using the following number line and table of rules:

6 9 2 8 3 1 4 0 5 7
L <————————————> R

1st digit left:	5 positions
2nd digit right:	2 positions
3rd digit right:	6 positions
4th digit left:	1 position

On the number line, locate 1 and move five positions to the left to get 6.
On the number line, locate 2 and move two positions to the right to get 3.
On the number line, locate 3 and move six positions to the right to get 6.
On the number line, locate 4 and move one position to the left to get 1.

E. Substitution with An Alphanumeric Line

Increase the substitution possibilities by using a line that includes both the alphabet as well as digits. For example, a ClientID of 1234 would be encrypted as I4PI using the following alphanumeric line and table of rules:

A B C D E F G H I J K L 0 1 2 3 4 5 6 7 8 9 M N O P Q R S T U V W X Y Z
L <————————————————————————————————> R

1st digit left:	5 positions
2nd digit right:	2 positions
3rd digit right:	10 positions
4th digit left:	8 positions

On the number line, locate 1 and move five positions to the left to get I.
On the number line, locate 2 and move two positions to the right to get 4.
On the number line, locate 3 and move ten positions to the right to get P.
On the number line, locate 4 and move eight position to the left to get I.

F. Key-Driven Substitution

A key can be used to drive the substitution algorithm. The key determines the number of positions to shift the characters of the ID. The selected key should be as long as the longest ID that exists. Each character of the key will correspond to one character of the ID.

The following example uses an algorithm that shifts every digit X positions to the right, where X is the alphabetic position of the corresponding letter of the key (i.e. A = 1, B = 2, C = 3, D = 4, E = 5, etc.). For example, if the number 123456789 is to be encrypted and the key is TRANSFORM, the resulting ID is XW4UAOYCZ using the alphanumeric line and table shown below:

A B C D E F G H I J K L 0 1 2 3 4 5 6 7 8 9 M N O P Q R S T U V W X Y Z
L <———————————————————————————————————> R

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

On the number, locate 1 and move (T= 20) positions to the right to get X.
On the number, locate 2 and move (R= 18) positions to the right to get W.
On the number, locate 3 and move (A= 1) position to the right to get 4.
On the number, locate 4 and move (N= 14) positions to the right to get U.
On the number, locate 5 and move (S= 19) positions to the right to get A.
On the number, locate 6 and move (F= 6) positions to the right to get O.
On the number, locate 7 and move (O= 15) positions to the right to get Y.
On the number, locate 8 and move (R= 18) positions to the right to get C.
On the number, locate 9 and move (M= 13) positions to the right to get Z.

IV. Transposition Method

A. Introduction

In this method of encryption, the positioning of the digits is changed. For example, 1234567 becomes 7234561 if the first and last digits are transposed. *Please note that this example is for illustration purposes only. It is not sufficiently complex for encrypting the title IV-E agency client ID.*

The following models (sections III.B. - III.C.) are appropriate for encrypting title IV-E agency client IDs.

B. Transposition of Digit Pairs

Transpose all digits in the ID. For example, 1234567 becomes 4657321, by applying the following rules:

Transpose the 1st and 7th digits Transpose the 2nd and 6th digits Transpose the 3rd and 5th digits Transpose the 1st and 4th digits

In the first transposition, 1 and 7 are switched to produce the number: 7234561.
In the second transposition, 2 and 6 are switched to produce the number: 7634521.
In the third transposition, 3 and 5 are switched to produce the number: 7654321.
In the fourth transposition, 7 and 4 are switched to produce the number: 4657321.

C. Transposition with Shuffled Digits

Rather than follow a straight transposition scheme, shuffle all digits. For example, 1234567 becomes 4715326 when applying the following rules:

1st digit to 3rd position 2nd digit to 6th position 3rd digit to 5th position 4th digit to 1st position 5th digit to 4th position 6th digit to 7th position 7th digit to 2nd position

In the first transposition, 1 will be in the third position: - - 1 - - - - .
In the second transposition, 2 will be in the sixth position: - - 1 - - 2 - .
In the third transposition, 3 will be in the fifth position: - - 1 - 3 2 - .
In the fourth transposition, 4 will be in the first position: 4 - 1 - 3 2 - .
In the fifth transposition, 5 will be in the fourth position: 4 - 1 5 3 2 - .
In the sixth transposition, 6 will be in the seventh position: 4 - 1 5 3 2 6 .
In the seventh transposition, 7 will be in the second position: 4 7 1 5 3 2 6 .

V. Bit-Level Operations

A. Introduction

Title IV-E agency case ID's can be encrypted by transforming the characters at the bit level to other characters in the ASCII data set. However, it is important to remember that ASCII control characters (0-31) and characters 35, 36, 37, and 64 must not be used. Inserting ASCII control characters into the AFCARS submission file will cause a file format error; the title IV-E agency file will not be processed and will fail the Data Compliance Utility.

Three methods of bit manipulation are left shifts, logical nots (binary complements), and exclusive or (XOR). Left shifts describe the procedure of re-positioning bits within a character. Logical nots describe an operation in which each bit is replaced with its complement. XOR describes an operation that involves comparing binary representations of two characters and producing another binary representation according to a set of comparison rules. Each bit-level operation is described below in greater detail. The following models (subsections IV.B. - IV.E.) are appropriate for encrypting a title IV-E agency's client IDs.

B. Left Shifts

This method has two types. Each involves three steps:

1. Truncate (delete) the left most bit.
2. Move the remaining bits over 1 position to the left.
   3.a. Fill the empty right position with a 0, or
3.b. Fill the empty right position with a 1.

B-1. Left Shift 1 Position

In this example, the left most bit is truncated, all of the remaining 0's and 1's are moved one position to the left, and the empty position filled with a 0:

Left Shift 1 position (<<1)			
	Binary Representation	ASCII number	ASCII Character
Original Character	00110011	51	3
Converted Character	01100110	102	f

B-2. Left Shift 1 Position Plus 1

In this example, the left most bit is truncated, all of the remaining 0's and 1's are moved one position to the left, and the empty position filled with a 1:

Left Shift 1 position and adding 1 (<<1) + 1			
	Binary Representation	ASCII number	ASCII Character
Original Number	00110011	51	3
Converted Character	01100111	103	g

C. Logical Nots (Binary Complements)

In this method, each bit is replaced with its complement (opposite value). Every 0 is replaced with a 1, every 1 is replaced with a 0. The following table illustrates this method:

Logical Not (~)			
	Binary Representation	ASCII number	ASCII Character
Original Number	00110011	51	3
Converted Character	11001100	204	╠

D. XOR Method

In this method, compare the binary representation of the original character and the binary representation of the key on a bit-by-bit basis. If both bits are the same (both 0 or both 1), the result is zero. If both bits are different, the result is one. The following table illustrates this method:

Exclusive or (XOR) method			
	Binary Representation	ASCII number	ASCII Character
Original Number	00110011	51	3
XOR Key	10000000	128	Ç
Final Conversion	10110011	179	│

E. Combinations of Left Shifting and Logical Nots

Bits can be further manipulated by sequentially applying two operations to a character. The following four tables illustrate some possible combinations:

E-1. Left Shift 1 Position Followed by a Logical Not

In this method, the left most bit is truncated, all of the remaining 0's and 1's are moved one position to the left, and a 0 is added to the right side. Next, each bit is replaced with its complement to get the final conversion shown below.

Left Shift 1 position (<<1) and a Logical Not (~)			
	Binary Representation	ASCII number	ASCII Character
Original Number	00110011	51	3
Initial Conversion	01100110	102	f
Final Conversion	10011001	153	Ö

E-2. Logical Not Followed by a Left Shift One Position

In this method, each bit is replaced with its complement. Next, the left most bit is truncated and all of the remaining 0's and 1's are moved one position to the left. A zero is added on the right side to get the final conversion shown below.

Logical Not (~) followed by a left shift 1 position (<<1)			
	Binary Representation	ASCII number	ASCII Character
Original Number	00110011	51	3
Initial Conversion	11001100	204	╠
Final Conversion	10011000	152	ÿ

E-3. Left Shift 1 Position Plus Followed by a Logical Not

In this method, the left most bit is truncated, all of the remaining 0's and 1's are moved one position to the left, and a 1 is added on the right side. Next, each bit is replaced with its complement to get the final conversion shown below.

Left Shift 1 position and adding 1 (<<1) + 1 and a Logical Not (~)			
	Binary Representation	ASCII number	ASCII Character
Original Number	00110011	51	3
Initial Conversion	01100111	103	g
Final Conversion	10011000	152	ÿ

E-4. Logical Not Followed by a Left Shift 1 Position Plus 1

In this method, each bit is replaced with its complement. Next, truncate the left most bit, all of the remaining 0's and 1's are moved one position to the left, and a 1 is added on the right side to get the final conversion shown below.

Logical Not (~) followed by a left shift 1 position and add 1 (<<1)+1			
	Binary Representation	ASCII number	ASCII Character
Original Number	00110011	51	3
Initial Conversion	11001100	204	╠
Final Conversion	10011001	153	Ö

VI. Combining Algorithms

A. Introduction

Title IV-E agencies can develop an encryption method that uses more the one algorithm. Combining methods can help create more complex encryption algorithms to ensure more secure title IV-E agency client IDs.

B. Substitution with Transposition

One possible combination is to apply substitution with shuffled digits to a title IV-E agency's client ID followed by transposition of digit pairs.

In the first method, 1234 is encrypted as 9715 using the following number line and table of rules:

9 0 8 2 3 6 7 4 1 5
L <———————————— > R

1st digit left:	8 positions
2nd digit right:	3 positions
3rd digit right:	4 positions
4th digit left:	8 positions

On the number line, locate 1 and move eight positions to the left to get 9.
On the number line, locate 2 and move three positions to the right to get 7.
On the number line, locate 3 and move four positions to the right to get 1.
On the number line, locate 4 and move eight positions to the left to get 5.

Next, transpose all digits in the ID. 9715 becomes 1597 by applying the following table of rules:

Transpose the 1st and 4th digits
Transpose the 2nd and 3rd digits
Transpose the 3rd and 4th digits
Transpose the 1st and 2nd digits

In the first transposition, 9 and 5 are switched to produce the number 5719.
In the second transposition, 7 and 1 are switched to produce the number 5179.
In the third transposition, 7 and 9 are switched to produce the number 5197.
In the fourth transposition, 5 and 1 are switched to produce the number 1597.

C. Bit Level Operations with Transposition

The following algorithm can be used by title IV-E agencies to create encrypted AFCARS record numbers and will ensure that none of the restricted ASCII codes (characters 0-31, 35, 36, 37, and 64) occur in the resulting encrypted numbers. It uses the operations described below and assumes that case IDs are composed of upper and lower case alphabetic characters (ASCII characters 41-90 and 97-122), digits 0-9 (ASCII characters 48-57), and dashes (ASCII character 45).

1. Set up a table assigning a value of 0 to 6 for the operations described below. Title IV-E agencies may use the following example table or modify the example by reordering the bit operations:

Numeric Assignment	Bit Operation	Description
0	<<1	1 Left Shift plus 0 bit
1	(<<1)+1	1 Left Shift plus 1 bit
2	~	Logical Not
3	<<1 ~	1 Left Shift plus 0 bit followed by a Logical Not
4	~ <<1	Logical Not followed by a 1 Left Shift plus 0 bit
5	(<<1)+1 ~	1 Left Shift plus 1 bit followed by a Logical Not
6	~ (<<1)+1	Logical Not followed by a 1 Left Shift plus 1 bit

2. Select an encryption key. The encryption key should be as long as the title IV-E agency case ID. The key should be kept secure and only be accessible to authorized State or Tribe personnel.

3. Apply the following steps to the case IDs:

    a. Determine the ASCII value of the first character of the key.

    b. Apply a MOD 7 operation to the ASCII value of that character. (In other words, divide the value by 7 and take the whole number remainder, which will be either 0,1,2,3,4,5, or 6).

    c. Using the chart developed in Step 1, go to the modal value (0-6) and select the corresponding bit operation.

    d. Determine the bit representation of the first character of the case ID.

e. Apply the bit operation(s) to the first character of the case ID number. The resulting character will be the first number encrypted.

f. Apply the same steps to each character in the ID, using the second character of the key to transform the second character of the case ID, the third for the third, etc.

The following is an illustration of these steps applied to the first letter of State case ID —BD-5983E25N5" using the key —Change code:"

a. "C" is the first letter of the key; its ASCII value is 67.

b. 67 MOD 7 = 4.

c. The corresponding bit operation is ~ <<1+0.

d. "B" is the first character of the State case ID. Its ASCII value is 66; the binary representation of 66 is 01000010.

e. Applying ~ <<1+0 to 01000010 will yield 01111010 or 122. 122 is the ASCII number for —z".

The following table details the conversion for each character of the ID:

Key	ASCII Value	MOD 7	Bit Operation	Case ID Character	ASCII Codes /Bit Values	Transformed ASCII Code /Bit Values	Encrypted Character
C	67	4	~ <<1	B	66/01000010	122/01111010	z
H	104	6	~ (<<1)+1	D	68/01000100	119/01110111	w
A	97	6	~ (<<1)+1	5	53/00110101	149/10010101	ò
N	110	5	(<<1)+1 ~	-	45/00101101	164/10100100	ñ
G	103	5	(<<1)+1 ~	9	57/00111001	140/10001100	î
E	101	3	<<1 ~	8	56/00111000	143/10001111	Å
<space>	32	4	~ <<1	3	51/00110011	152/10011000	ÿ
c	99	1	(<<1)+1	E	69/01000101	139/10001011	Ï
o	111	6	~ (<<1)+1	2	50/00110010	155/10011011	¢
d	100	2	~	5	53/00110101	202/11001010	╩
e	101	3	<<1 ~	N	78/01001110	99/01100011	C
.	46	4	~ <<1	5	53/00110101	148/10010100	Ö

We will now change the position of the encrypted characters using the following table of rules:

Transpose the 1st and 12th character
Transpose the 2nd and 11th character
Transpose the 3rd and 10th character
Transpose the 4th and 9th character
Transpose the 5th and 8th character
Transpose the 6th and 7th character

In the first transposition, z and ö are switched to produce the character set:

ö w ò ñ î Åy ï ¢ ╩ c z

In the second transposition, w and c are switched to produce the character set:

ö c ò ñ î Åy ï ¢ ╩ w z

In the third transposition, ò and ╩ are switched to produce the character set:

ö c ╩ ð î Åy ï ¢ ò w z

In the fourth transposition, ñ and ¢ are switched to produce the character set:

ö c ╩ ¢ î Åy ï ñ ò w z

In the fifth transposition, î and ï are switched to produce the character set:

ö c ╩ ¢ ï Åy î ñ ò w z

In the sixth transposition, Å andy are switched to produce the final character set:
ö c ╩ ¢ ïy Å î ñ ò w z

The final character set now represents the unique encrypted character set.

VII. Reversing The Encryption Algorithm

A. Introduction

As mentioned in the *Guidelines (Section B above)*, be sure that the selected encryption method can be reversed to produce the original client ID for any AFCARS record number.

B. Sample Reversal Algorithm

We will use the algorithm described in section V.B. (substitution with transposition) to illustrate the process. Let us work backwards starting with the second encryption method (transposition of digit pairs). Our first step is to reverse the order of the digit pairs that were transposed to give us 1597. With each digit pair, reverse the switching order to produce the following table of rules:

Transpose the 2nd and 1st digits
Transpose the 4th and 3rd digits
Transpose the 3rd and 2nd digits
Transpose the 4th and 1st digits

In the first transposition, 5 and 1 are switched to produce the number 5197.
In the second transposition, 7 and 9 are switched to produce the number 5179.
In the third transposition, 7 and 1 are switched to produce the number 5719.
In the fourth transposition, 5 and 9 are switched to produce the number 9715.

We will now change 9715 back to the original number (1234) by reversing the original substitution method. The number line will not change. Using the original table of rules, reverse the direction for each rule to produce the following table of rules:

9 0 8 2 3 6 7 4 1 5
L <—————————————— > R

1st digit right:	8 positions
2nd digit left:	3 positions
3rd digit left:	4 positions
4th digit right:	8 positions

On the number line, locate 9 and move eight positions to the right to get 1.
On the number line, locate 7 and move three positions to the left to get 2.
On the number line, locate 1 and move four positions to the left to get 3.
On the number line, locate 5 and move eight positions to the right to get 4.

As stated in the *Guidelines*, be sure that the encrypted number is scanned so that the ASCII control numbers (0-31) and ASCII characters 35, 36, 37, and 64 are not being used.

Bibliography

The sources below have examples of additional encryption strategies and techniques.

Denning, Dorothy E.R., *Cryptography and Data Security,* Addison Wesley, 1983.

Frank, Peter, *Calculator Ciphers,* Information Associates, 1980.

Kahn, David, *The Codebreakers,* Macmillan Company, 1972.

Marotta, Michael E., The Code Book: *All about Unbreakable Codes and How to Use Them,* Loompanics Unlimited, 1987

Meyer, Carl H. and Matyas, Stephen M., *Cryptography,* John Wiley & Sons, 1982.

Prosise, Jeff, *How to Keep it a Secret,* PC Magazine, July 1994.

Prosise, Jeff, *WinCrypt Protects Your Data,* PC Magazine, July 1994.

Russell, Deborah and Gangemi, G.T., *Computer Security Basics,* O'Reilly & Associates, Inc., 1992.