Paper 3880-2019

Cryptosystem for Protecting Personal Information and Data Visualization using SAS® Visual Analytics

Patrick Sekgoka, South African Reserve Bank; Olufemi Adetunji, University of Pretoria

ABSTRACT

Data comprising private and confidential information is often stored with multiple records per subject in a variety of fields. It is important to know how to process such data sets with grouped observations while complying with information privacy laws and policies. In this paper, we demonstrate how to use temporary automatic variables such as _N_, First.BY-variable and Last.BY-variable from the SAS® Program Data Vector (PDV) to implement a cryptosystem for protection of personally identifiable information in a data set comprising cross-border financial flows. We also construct a bipartite graph using a network diagram in SAS® Visual Analytics for SAS®9 to visualize the cross-border financial flows data set.

INTRODUCTION

The DATA step is the primary method for creating a SAS® data set. A good understanding of DATA step concepts such as DATA step processing, reading raw data, BY-group processing, combining and modifying data sets among others, allows one to benefit from the SAS® software investment.

In this paper, we use BY-group processing to implement a cryptosystem for protection of personally identifiable information. BY-group processing is commonly used in the DATA step to combine two or more data sets using a BY statement with a SET, MERGE, MODIFY, or UPDATE statement. Hence, it provides users with efficient ways to navigate through the data sets with one or more grouping variables.

We use a network diagram depicted in Figure 1 to visualize the relationships among the subjects between two groups. This network diagram is called a bipartite network. It comprises two disjoint sets of nodes that are differentiated by their colors with the condition that no two connected nodes are of the same color. In this example data set, the two sets of nodes represent “Residents” and “Non-residents” in cross-border financial transactions denoted by “R” and “NR”, respectively.

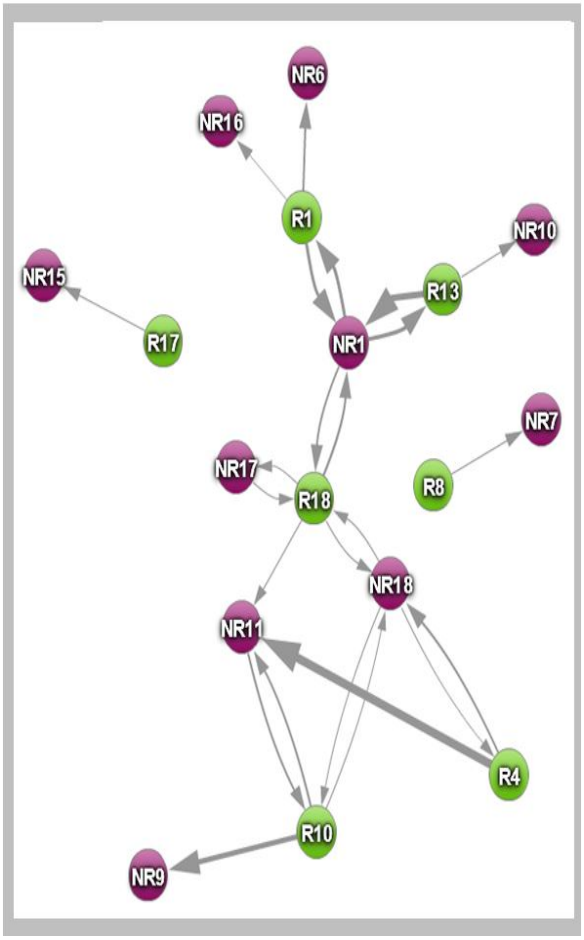

Figure 1: Bipartite Network Diagram for visualization of cross-border financial flows.

In the next section, we provide some background about concerns that often arise when one is tasked with analyzing data containing personally identifiable information, followed by the implementation of a simple cryptosystem that uses BY-group processing in the DATA step. Thereafter, we create a visualization of the cross-border financial flows data set and conclude.

INFORMATION PRIVACY AND DATA ENCRYPTION

Data extracts from a wide range of sources such as financial transactions held by financial institutions, patient records held by healthcare system (providers), salary records held by employers, investigation records held by the criminal justice system, motor vehicle registration information held by public institutions, etc., often trigger information privacy concerns whenever analytical tasks are carried out on such data sets. Hence, information privacy has emerged as one of major concerns for governments, firms and individuals. Compliance with data protection regulations by private and public firms is required whenever one processes data comprising of personally identifiable information. We adopt a cryptographic approach to addressing the information privacy concerns.

We use the example data set (transactions.xlsx) in Figure 2 to implement the cryptosystem in SAS®. Each transaction is a record of cash received/paid by a resident individual/firm of a country from/to a non-resident individual/firm. The names of residents and non-residents are regarded as personally identifiable information. Such data sets are primarily held by licensed foreign exchange dealers, commercial and central banks, among others.

Resident name	Transaction date	Flow	NonResident name	Amount
Lynn	1/16/2018	Out	Joaquin	3500
Elizabeth	1/25/2018	Out	Victoria	1000
Christo	2/7/2018	Out	Mavis	500
Elizabeth	2/14/2018	In	Victoria	250
Lynn	3/4/2018	In	Victoria	100
Rosalia	3/12/2018	In	Sara	200
Christo	3/12/2018	In	Benjamin	2500
Martina	3/30/2018	In	Benjamin	1000
Martina	3/31/2018	Out	Benjamin	2250
Rosalia	4/18/2018	Out	Mariana	250
Martina	4/23/2018	In	Benjamin	1250
Rosalia	5/5/2018	In	Benjamin	1000
Christo	6/5/2018	Out	Catalina	1000
Michael	6/10/2018	Out	Matias	500
Elizabeth	6/15/2018	Out	Mariana	1000
Elizabeth	6/18/2018	Out	Mariana	5000
Rosalia	7/2/2018	In	Victoria	150
Linda	7/11/2018	Out	Diego	275
Lynn	7/14/2018	In	Mariana	750
Linda	8/8/2018	Out	Diego	275
Martina	8/17/2018	Out	Lucas	500
Rosalia	9/2/2018	Out	Victoria	150

Figure 2: Cross-border transactions data (transactions.xlsx) – the example data set.

The goal is to replace the personally identifiable information with labels prior to analyzing the data set. To achieve this goal, we start by importing the data set from MS Excel into SAS® using the IMPORT procedure and proceed to encrypt the data set in the DATA step.

Thereafter, we prepare the resulting data set for input into Visual Analytics. The code for importing the data is as follows:

```
proc import out = ExampleData datafile = "path\transactions.xlsx"
   dbms = xlsx replace;
   sheet = "Data";
   getnames = yes;
run;
```

The DATA step uses the BY statement along with a SET statement. Hence, we first sort the data set by the BY variable. The next SORT procedure sorts the data set by both BY variables before encryption of the variable “Resident_name” in the DATA step. The code is as follows:

```
proc sort data = ExampleData;
  by Resident_name NonResident_name;
run;

data CrossBorderData;
  set ExampleData;
   by Resident_name;
   retain resident label;
     if FIRST.Resident_name then
     Resident label = CAT('R',_N_);
run;
```

During the first iteration of the DATA step above, SAS® processes the first observation of the data set, setting the value of _N_ to 1. The BY statement used along with the SET statement instructs SAS® to create two automatic variables, i.e., First.Resident_name and Last.Resident_name in the PDV. Since Christo is the first resident name in the data set, First.Resident_name is set to the value of 1. SAS® looks ahead at the next observation to determine Last.Resident_name. This is set to zero since the second observation is the same as the first. Figure 3a shows the contents of the two automatic variables in the PDV during the first iteration of the DATA step. The subsetting IF statement causes the DATA step to continue processing only Christo’s record by concatenating the letter “R” with the automatic variable _N_ to result with a resident label “R1”.

Resident_name	Transaction_date	Flow	NonResident_name	Amount
Christo	2018-02-07	Out	Mavis	500
Christo	2018-03-12	In	Benjamin	2500
Christo	2018-06-05	Out	Catalina	1000
Elizabeth	2018-01-25	Out	Victoria	1000
Elizabeth	2018-02-14	In	Victoria	250
Elizabeth	2018-06-15	Out	Mariana	1000
Elizabeth	2018-06-18	Out	Mariana	5000

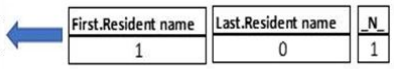

Figure 3a: PDV contents of First. / Last. Resident_name values during the first iteration.

The RETAIN statement, which is a compile-time-only statement retains the value of “Resident_label” in the PDV across iterations of the DATA step. It is quite critical to use the RETAIN statement since the PDV variables are reinitialized at every iteration of the DATA step. The contents of the automatic variables during the next three iterations of the DATA step are shown in Figure 3b to Figure 3d.

Resident_name	Transaction_date	Flow	NonResident_name	Amount
Christo	2018-02-07	Out	Mavis	500
Christo	2018-03-12	In	Benjamin	2500
Christo	2018-06-05	Out	Catalina	1000
Elizabeth	2018-01-25	Out	Victoria	1000
Elizabeth	2018-02-14	In	Victoria	250
Elizabeth	2018-06-15	Out	Mariana	1000
Elizabeth	2018-06-18	Out	Mariana	5000

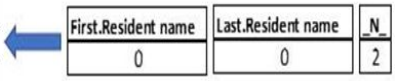

Figure 3b: PDV contents of First. / Last. Resident_name values during the second iteration.

Resident_name	Transaction_date	Flow	NonResident_name	Amount
Christo	2018-02-07	Out	Mavis	500
Christo	2018-03-12	In	Benjamin	2500
Christo	2018-06-05	Out	Catalina	1000
Elizabeth	2018-01-25	Out	Victoria	1000
Elizabeth	2018-02-14	In	Victoria	250
Elizabeth	2018-06-15	Out	Mariana	1000
Elizabeth	2018-06-18	Out	Mariana	5000

Figure 3c: PDV contents of First. / Last. Resident_name values during the third iteration.

Resident_name	Transaction_date	Flow	NonResident_name	Amount
Christo	2018-02-07	Out	Mavis	500
Christo	2018-03-12	In	Benjamin	2500
Christo	2018-06-05	Out	Catalina	1000
Elizabeth	2018-01-25	Out	Victoria	1000
Elizabeth	2018-02-14	In	Victoria	250
Elizabeth	2018-06-15	Out	Mariana	1000
Elizabeth	2018-06-18	Out	Mariana	5000

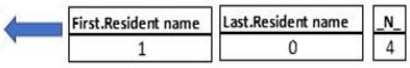

Figure 3d: PDV contents of First. / Last. Resident_name values during the fourth iteration.

Note the change of resident name from Christo to Elizabeth during the fourth iteration of the DATA step in Figure 3d above. At this step, First.Resident_name is set to the value of 1 to indicate the start of DATA step processing for the resident name Elizabeth. SAS® looks ahead at the next observation to determine Last.Resident_name, which is set to zero next resident name is also Elizabeth. The encryption process in the DATA step is continued until full encryption of both resident and non-resident names is achieved.

The PRINT procedure below gives the partial printout of the output data set after completion of the DATA step.

```
proc print data = CrossBorderData (obs = 15);
run;
```

Obs	Resident_name	Transaction_date	Flow	NonResident_name	Amount	Resident_label
1	Christo	3/12/2018	In	Benjamin	2500	R1
2	Martina	3/30/2018	In	Benjamin	1000	R13
3	Martina	3/31/2018	Out	Benjamin	2250	R13
4	Martina	4/23/2018	In	Benjamin	1250	R13
5	Rosalia	5/5/2018	In	Benjamin	1000	R18
6	Christo	6/5/2018	Out	Catalina	1000	R1
7	Linda	7/11/2018	Out	Diego	275	R8
8	Linda	8/8/2018	Out	Diego	275	R8
9	Lynn	1/16/2018	Out	Joaquin	3500	R10
10	Martina	8/17/2018	Out	Lucas	500	R13
11	Elizabeth	6/15/2018	Out	Mariana	1000	R4
12	Elizabeth	6/18/2018	Out	Mariana	5000	R4
13	Lynn	7/14/2018	In	Mariana	750	R10
14	Rosalia	4/18/2018	Out	Mariana	250	R18
15	Michael	6/10/2018	Out	Matias	500	R17

Figure 4: The example data set with a new variable "Resident_label" for each resident name.

Next we repeat the SORT procedure and DATA step process to create the label for non-resident names to complete the encryption process. The code for completing the encryption process is as follows:

```
proc sort data = CrossBorderData;
  by NonResident_name Resident_name;
run;

data CrossBorderDataFinal;
   set CrossBorderData;
    by NonResident_name;
    retain NonResident_label;
    if FIRST.NonResident_Name then
    NonResident_label = CAT('NR',_N_);
run;
```

The output data set from the DATA step contains the two additional variables created during the encryption process. The partial printout of the output data is shown in Figure 5. The data set in Figure 5 serves as the key for the cryptosystem and must be stored in a secure medium with restrictions.

Obs	Resident_name	Transaction_date	Flow	NonResident_name	Amount	Resident_label	NonResident_label
1	Christo	3/12/2018	In	Benjamin	2500	R1	NR1
2	Martina	3/30/2018	In	Benjamin	1000	R13	NR1
3	Martina	3/31/2018	Out	Benjamin	2250	R13	NR1
4	Martina	4/23/2018	In	Benjamin	1250	R13	NR1
5	Rosalia	5/5/2018	In	Benjamin	1000	R18	NR1
6	Christo	6/5/2018	Out	Catalina	1000	R1	NR6
7	Linda	7/11/2018	Out	Diego	275	R8	NR7
8	Linda	8/8/2018	Out	Diego	275	R8	NR7
9	Lynn	1/16/2018	Out	Joaquin	3500	R10	NR9
10	Martina	8/17/2018	Out	Lucas	500	R13	NR10
11	Elizabeth	6/15/2018	Out	Mariana	1000	R4	NR11
12	Elizabeth	6/18/2018	Out	Mariana	5000	R4	NR11
13	Lynn	7/14/2018	In	Mariana	750	R10	NR11
14	Rosalia	4/18/2018	Out	Mariana	250	R18	NR11
15	Michael	6/10/2018	Out	Matias	500	R17	NR15

Figure 5: Financial transactions data set with two additional variables "Resident_label" and

"NonResident_label" to represent the identity of residents and non-residents on the data set.

To obtain the final data set, we use the DROP= data set option to specify the variables to exclude from the last output data set in Figure 5. Thus, the encryption process ends with a data set that does not contain personally identifiable information. This data set can be made available for analytical purpose within a firm with very limited security restrictions to encourage data discovery. The code for creating this data set is as follows:

```
data CrossBorderAnalysis;
  retain Resident_label Flow Amount NonResident_label;
  set CrossBorderDataFinal (drop = Transaction_date Resident_name
  NonResident_name);
run;
```

The original data set in Figure 2 contained personally identifiable information, which we refer to as plaintexts. We transformed the plaintexts into cyphertexts using the DATA step to result with the encrypted data set, which is partially is shown in Figure 6.

Obs	Resident_label	Flow	Amount	NonResident_label
1	R1	In	2500	NR1
2	R13	In	1000	NR1
3	R13	Out	2250	NR1
4	R13	In	1250	NR1
5	R18	In	1000	NR1
6	R1	Out	1000	NR6
7	R8	Out	275	NR7
8	R8	Out	275	NR7
9	R10	Out	3500	NR9
10	R13	Out	500	NR10
11	R4	Out	1000	NR11
12	R4	Out	5000	NR11
13	R10	In	750	NR11
14	R18	Out	250	NR11
15	R17	Out	500	NR15

Figure 6: Encrypted cross-border transactions data.

VISUALIZATION OF CROSS_BORDER TRANSACTIONS USING A NETWORK DIAGRAM

In this section we create a visual display of the encrypted data set using a network diagram in the SAS® Visual Analytics Explorer window. We use the "ungrouped" node-link pairs, which require data to be structured to fit the basic data roles being "Source" and "Target". Payment flows from residents to non-residents have residents as source nodes and non-residents as target nodes whereas the payment flows from non-residents to residents will have non-residents as source nodes and residents as target nodes. In addition, we introduce a binary variable (resident indicator) to the data set in order to keep track of whether a node is a resident node or non-resident node to ensure that the bipartite structure of the network is depicted. The resulting data set shown in Figure 7 is loaded into the SAS LASR server and used to create a network diagram with the following roles:

- Network type = Ungrouped
- Source = Source Node (determined by the direction of payment flows between residents and non-residents)
- Target = Target Node (determined by the direction of payment flows between residents and non-

residents)

- Node size empty
- Node color = Resident indicator (binary 1 = yes, 0 = no)
- Link width = Amount (thick links for large payment flows)
- Link color and data tips empty.

Figure 8 shows the roles as displayed in the SAS® Visual Analytics Explorer window. The node colors and node labels are selected using the network properties tab in the Visual Analytics Explorer window. The resulting bipartite network is shown in Figure 1.

Source Node	Target Node	Amount	Resident Indicator
R1	NR1	2500	1
R13	NR1	1000	1
R13	NR1	1250	1
R18	NR1	1000	1
R10	NR11	750	1
R18	NR17	200	1
R4	NR18	250	1
R10	NR18	100	1
R18	NR18	150	1
R13	NR1	2250	1
R1	NR6	1000	1
R8	NR7	275	1
R8	NR7	275	1
R10	NR9	3500	1
R13	NR10	500	1
R4	NR11	1000	1
R4	NR11	5000	1
R18	NR11	250	1
R17	NR15	500	1
R1	NR16	500	1
R4	NR18	1000	1
R18	NR18	150	1
NR1	R1	2500	0
NR1	R13	1000	0
NR1	R13	1250	0
NR1	R18	1000	0
NR11	R10	750	0
NR17	R18	200	0
NR18	R4	250	0
NR18	R10	100	0
NR18	R18	150	0
NR6			0
NR7			0
NR7			0
NR9			0
NR10			0
NR15			0
NR16			0

Figure 7: Input data set prepared for creation of a network diagram using the ungrouped network type.

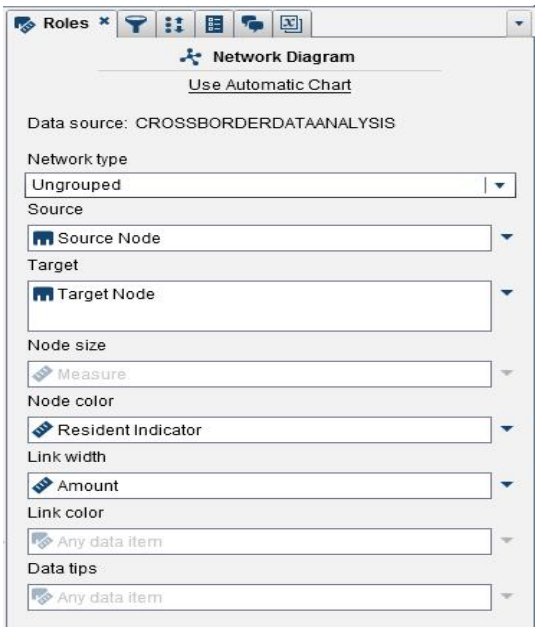

Figure 8: Network diagram roles.

CONCLUSION

The protection of private and confidential information is an important issue. We adopted a cryptographic approach to addressing this issue using the flexibility of the DATA step in SAS®. The implementation of the cryptosystem has shown that the use of temporary automatic variables from the PDV enables efficient processing of data sets in the presence of one or more grouping variables. Visualization of real world data sets with similar characteristics to the data considered in this paper can be done using SAS® Visual Analytics.

RECOMMENDED READING

- *SAS® Programming 2: Data manipulation Techniques (Course Notes)*
- *SAS®9.2 Language Reference: Concepts, Second Edition: By-Group Processing in the DATA step*
- *SAS® Visual Analytics for SAS®9*
- *Visualizing Relationships and Connections in Complex Data Using Network Diagrams in SAS® Visual Analytics, Stephen Overton, Ben Zenick, Zencos Consulting, Paper 3323-2015*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Patrick Sekgoka
Financial Surveillance Department
South African Reserve Bank
Patrick.Sekgoka@resbank.co.za

Olufemi Adetunji
Department of Industrial and Systems Engineering
University of Pretoria
Olufemi.Adetunji@up.ac.za