

# RDF XML Customized Protocol (RXCP)

Siva Srinivasa Rao Mothukuri

PG Student

Department of Information Technology  
, RVR & JC, Acharya Nagarjuna University  
Guntur, Andhra Pradesh, India.

## Abstract

Digital data is increasing exponentially with the use of computational electronic devices, resulted in marking data with different types of formats. When it comes to the handling of the data by other end user, it becomes tedious as multiple formats of data will be received and possibly tampered data. To absolve this, protocol was introduced, providing set of rules. This paper provides a new protocol for RDF and XML data that need to be transferred over the network. In addition to this, there are two more features added to the protocol like customization which provides end user to set customized rules for incoming data that need to be processed and other is security. Security can be customized based on the requirement of the end user by invoking different types of security algorithms and keys with ease use.

**Keywords:** Protocol, TCP/IP, RDF, XML, Cryptography, DES, AES

## I. INTRODUCTION

Today's world, every computational device is connected to another through internet like PC, Mobiles, and Tabs and so on. There comes the question, how the data is transferred from one to another. The simple word that gives answer is TCP/IP. Even though there are other variants of architectures, TCP/IP has been adapted globally. TCP/IP takes all the requests and sends it to upper layers i.e., to end users. This is controlled using other protocols namely FTP, SSH, SNMP, SMTP and so on. On the other side of the coin, data sent over the network is simply a raw data which has no rules. This can be converted into organized structure by XML or RDF. By transforming data into XML or RDF file, there can be specific rules for every data that is transferred. Adding both components gives rise to new protocol that is introduced in this paper (RXCP). Furthermore, adding a customization gives the end user to transfer particular type of data with particular fields specified in the rules. In this work, we introduce the concept of packet attribute, which is used to identify the packets from different applications or heterogeneous sensors according to specific requirements, then design an attribute-aware data aggregation (ADA) scheme, which can make the packets with the same attribute convergent as much as possible to improve the efficiency of data aggregation. A distributed and dynamic routing protocol is expected to adapt to the frequent variation of packet attribute distribution at each node.

## II. RDF XML CUSTOMIZATION PROTOCOL (RXCP)

RXCP is designed debating three features; socket, customization, security. Socket is a simple TCP/IP connection oriented socket connection accepting all incoming packets and processing. Customization is achieved by having a set of rules that must be included by the end user. Finally, Security is an additional add-on as the data can't be sent over the network

### A. Socket:

RXCP is implemented on top of TCP/IP. So, it acts as similar other protocols like FTP or SSH. As we know for a protocol to be implemented there must be a static dedicated port. Static port "9812" is taken and RXCP is built using the port

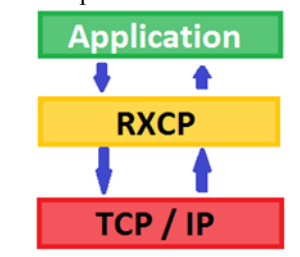


Fig. 1: A. Socket

The data is sent over the network in the form of bytes. Final data is passed through this socket after completing security and processing stage. At the server side, socket waits for the incoming data. When a new data is received, then it is passed to next stage for processing. For closing of the connection, a simple handshake is performed. Further, server or client closes the

connection. As the data is encrypted by the end application user, there is a negligible attack performed on the data over the network. Client initiates the connection using socket connection, thereby handovers the data to server.

### B. Customization:

Protocol is a set of rules that must be followed in-order to maintain accurate and coherent information. RXCP provides a distinct set of rules for every xml/rdf file. However, xml/rdf is user defined. So, the rules must be user defined which leads to customization. User can define multiple set of rules for data. Depending on the rules, the data is accepted or dropped.

- Following sample xml file is generated using a form.

<!--Sample XML file generated From form

Author::Siva Srinivasa Rao Mothukuri -->

<?xml version="1.0"?>

<book>

<chapter>

<page>XML</page>

<page>RDF</page>

<page>Protocol</page>

<note value="note1" />

<note value="note2" />

<note value="note3" />

<note value="note4" />

</chapter>

<chapter>

<page>XML1</page>

<page>RDF2</page>

<page>Protocol3</page>

<note value="note11" />

<note value="note22" />

<note value="note33" />

<note value="note44" />

</chapter>

</book>

- Following sample ini file contains rules.

1-book [s]

1-1-chapter [m]

1-1-1-page [3]

1-1-2-note [sem]

- Sample RDF File

<?xml version="1.0"?>

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description

rdf:about="http://www.recshop.fake/cd/Empire Burlesque">

<cd:artist>Bob Dylan</cd:artist>

<cd:country>USA</cd:country>

<cd:company>Columbia</cd:company>

<cd:price>10.90</cd:price>

<cd:year>1985</cd:year>

</rdf:Description>

<rdf:Description

rdf:about="http://www.recshop.fake/cd/Hide your heart">

<cd:artist>Bonnie Tyler</cd:artist>

<cd:country>UK</cd:country>

<cd:company>CBS Records</cd:company>

<cd:price>9.90</cd:price>

<cd:year>1988</cd:year>

</rdf:Description>

</rdf:RDF>

- Sample ini file for RDF

1-rdf:RDF [S]  
1-1-rdf:Description [2]  
1-1-1-cd:artist [S]  
1-1-2-cd:country [S]  
1-1-3-cd:price [S]  
1-1-4-cd:year [1]

Following table contains the operations that can be included in ini file.

Table-1

| Notation   | Description                                         |
|------------|-----------------------------------------------------|
| [1..n]     | Opening or primary tags that are involved           |
| 1-[1..n]   | Sub-tag that can be included with numbering         |
| S          | Single tag (If exceeds then data will be discarded) |
| M          | Multiple tags                                       |
| [{1..n}]   | Count for particular tag                            |
| SE         | Self-Ending/Single-Ending tag                       |
| SE[{1..n}] | Self-Ending Tag with count                          |
| SEM        | Self-Ending Tag with multiple count                 |

Note there is difference between Single tag and count 1, if the user specifies as Single tag, it is mandatory to have the specified tag. If it is specified as 1, then the tag is optional. This is valid only for Single and count 1.

### C. Security:

As already mentioned above, the security is an add-on provided for encrypting the data over the network. This add-on is provided to end user who can configure required security settings in server and client side. With a simple change user can change the encryption algorithm.

– Sample Properties file

# Defining Keys

```
key1=g3HkeS'[u,`bKR,FDh>.3?rE.9[4>'mf^X=b'_Rr(kQf.6JASsL=hvkW>h>GcFbG;=m:!msW:WJqHf@G$6ZvPj-{a)BP<mz,q
key2=tu$g]a(j^^;G<`VL]xD`eASyk2G"%84>j`X5)Gt`b8k8r9;p(PNdHC[hGmWCWAK:VdEqE.K!D,<j#)s4a(g.5pb7k78W^tmy_
{s
```

# Security Algorithm

Security=DES

# Available Security Algorithms -- DES, AES, TRIPLE DES, NO-SECURITY.

Key1 and Key2 are parameters that take user defined values and encrypt the data based on those keys. Security is another parameter that takes type of algorithm like DES, AES, Triple DES and NO-SECURITY. Based on the input, the encryption algorithm was taken. For triple DES, the encryption is done with provided two keys along with another which is hardcoded in the protocol.



Fig. 2: Security Algorithms

## III. IMPLEMENTATION

The implementation of the RXCP is done based on the parsers available in hand i.e., SAX Parser. Whenever the data is received on the server side, it passes the data to decrypter. After completing decryption phase, the data is sent to parser for parsing the data. Along with parsing, validations are processed. Finally, accepted data is used as per requirement such as inserting into database or showing it to other users.

Following are the sample snapshots of implementation of RXCP

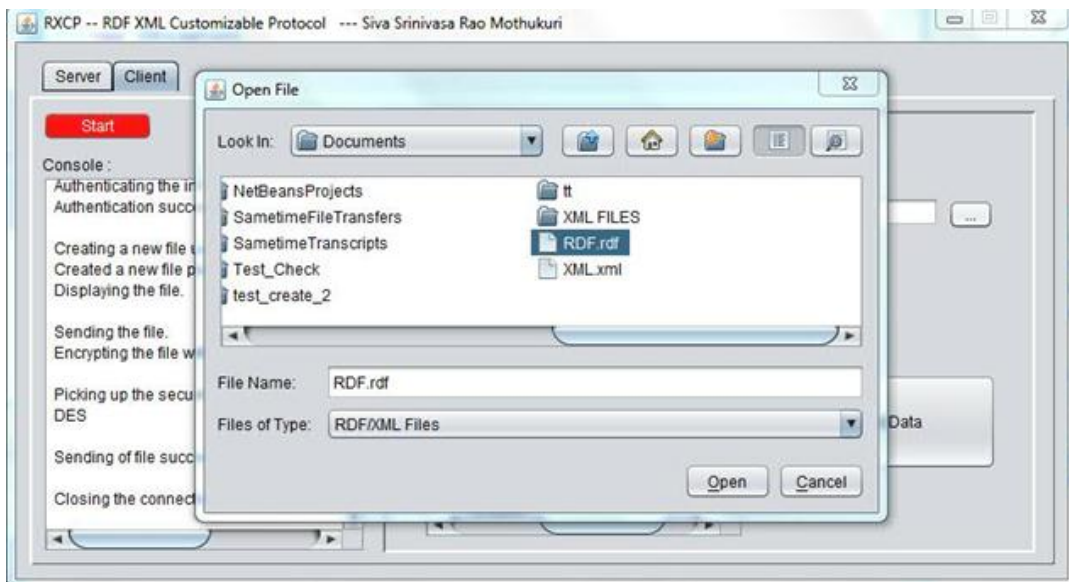


Fig. 3: Shows Selection of RDF/XML File for Sending Over the Network

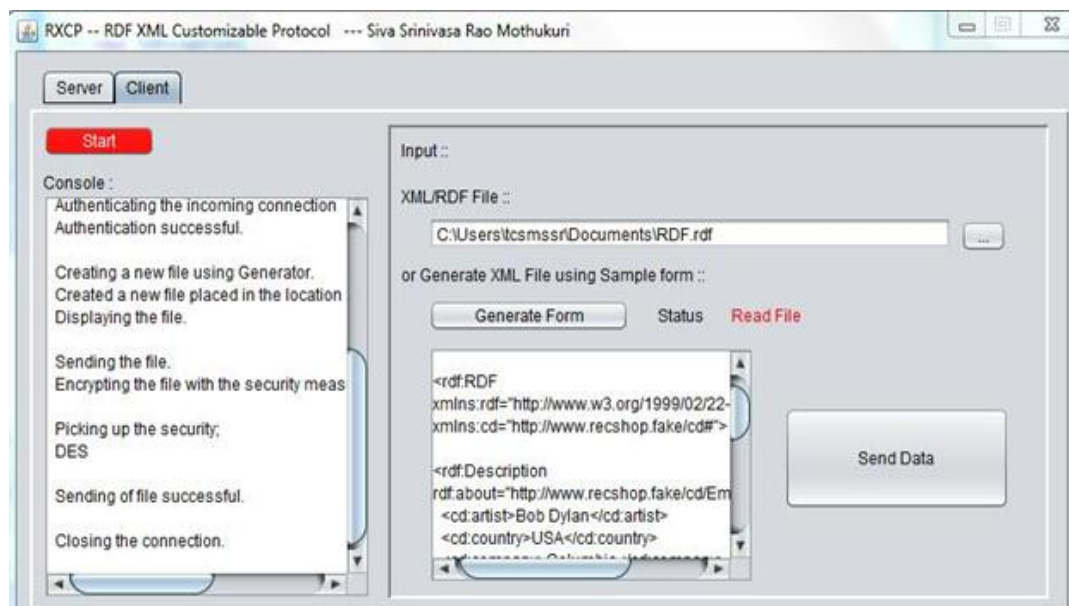


Fig. 4: Snapshot Show that Imported Data and Status in Application.

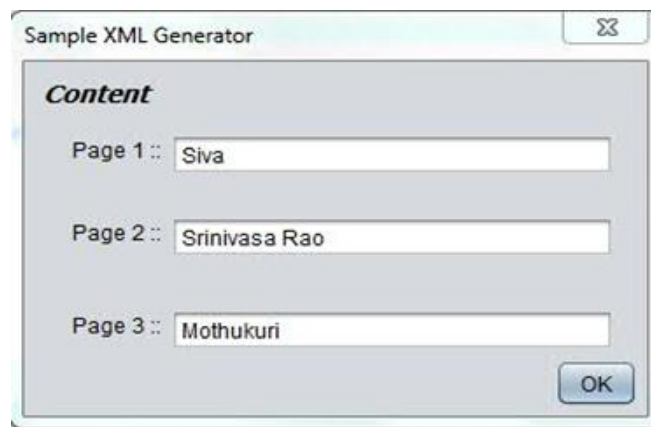


Fig. 5: Shows A Form For Generating Xml File Using Sample Data

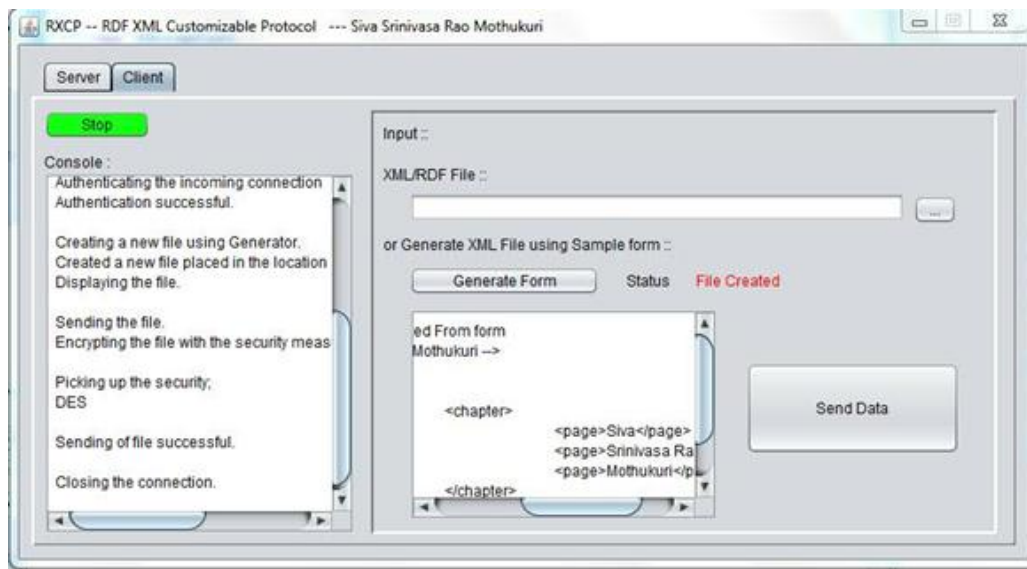


Fig. 6: Shows Xml Data In The Application And Ready For Sending It.

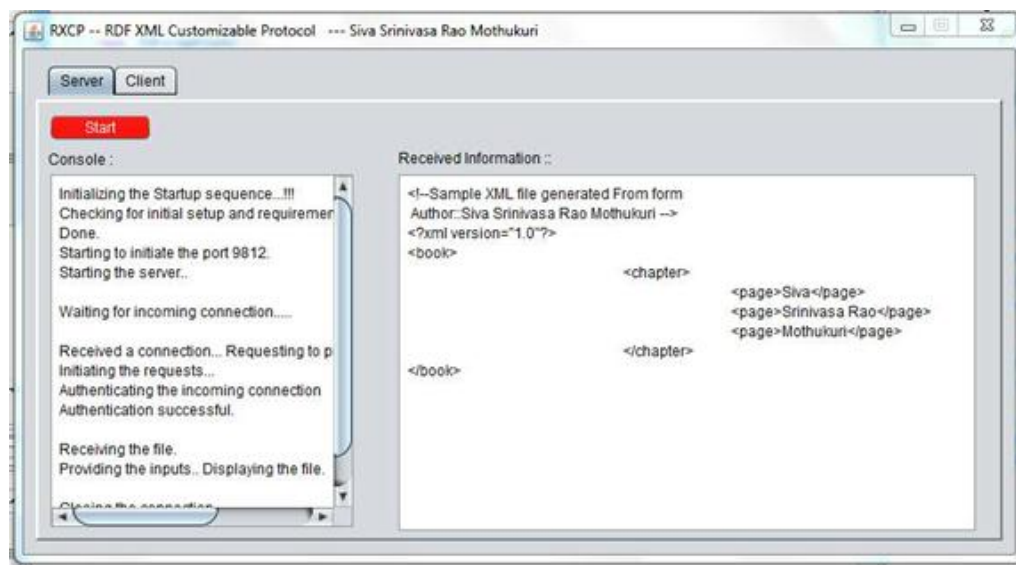


Fig. 7: Shows That Server Side Has Received The Data.

## IV. CONCLUSION

Every data sent over the network can't be accepted by the server for processing. There must a set of rules that must be followed for every data to process and discard unnecessary. This gives a new way to have RDF/XML data with customization for every tag that is present in the data which offers a great control over it. By having security as another feature, provides classification of data over the network, resulting in reduced attacks on the data.

## REFERENCES

- [1] Case, J. D., Fedor, M., and Schoffstall, M. L., "Simple Network Management Protocol(SNMP)." Internet Request for Comments 1157, May 1990.
- [2] Comer, Douglas E., Internetworking with TCP/IP, Volume k Principles, Protocols, andArchitecture (third edition). Prentice Hall, 1995.
- [3] Comer, Douglas E., and Stevens, David L., Internetworking with TCP/IP, Volume II: Design,Implementation, and Internals (third edition). Prentice Hall, 1999.
- [4] Comer, Douglas E., and Stevens, David L., Internetworking with TCP/IP, Volume III: ClientServerProgramming and Applications (BSD version, second edition). Prentice Hall, 1996.
- [5] Deering, S., and Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification." Internet Request for Comments 2460, December 1998.
- [6] Gilligan, R., Thomson, S., Bound, J., and Stevens, W., "Basic Socket Interface Extensions for IPv6." Internet Request for Comments 2553, March 1999.
- [7] Hughes, M., Shoffner, M., Hamner, D., and Bellus, U., Java Network Programming (second edition). Manning, 1999.
- [8] International Organization for Standardization, Information Processing Systems Open Systems InterconnectionmSpecification of Abstract Syntax Notation One (ASN.1). International Standard 8824, December 1987.
- [9] Mockapetris, P., "Domain Names--Concepts and Facilities." Internet Request for Comments 1034, November 1987.

- [10] Mockapetris, P., "Domain Names--Implementation and Specification." Internet Request for Comments 1035, November 987.
- [11] Peterson, L. L., and Davie, B. S., Computer Networks: A Systems Approach (second edition). Morgan Kaufmann, 2000.
- [12] Postel, J., "Internet Protocol." Internet Request for Comments 791, September 1981.