Notes on Intuitionistic Fuzzy Sets
ISSN 1310–4926
Vol. 20, 2014, No. 3, 65–71

Intuitionistic fuzzy estimations of establishing secure File Transfer Protocol connection with Transport Layer Security

Ivelina Vardeva

"Prof. Asen Zlatarov" University
1 "Prof. Yakimov" Blvd., Burgas 8000, Bulgaria
e-mail: ivardeva@gmail.com

Abstract: The present article describes a method for calculating intuitionistic fuzzy estimations of establishing secure FTP connection with TLS. The model description and analysis is done in the terms of generalized nets.
Keywords: Intuitionistic fuzzy estimations (IFEs), Generalized nets (GNs), File Transfer Protocol (FTP), Transport Layer Security (TLS).
AMS Classification: 03E72.

1 Introduction

The Transport Layer Security (TLS) protocol is a popular mechanism for generally securing a socket connection. The security extensions to FTP, presented in [8], offer a comprehensive set of commands and responses that can be used to add authentication, integrity, and confidentiality to the FTP protocol.

TLS is a protocol that synchronizes multiple transactions (handshakes) at a time, supporting large numbers of users connecting a server. This requires that the TLS protocol be described in terms of a mathematical tool for modelling that supports multiple parallel processes and competitive behaviour, and such a tool for modelling is the herewith chosen apparatus of generalized nets.

Generalized nets (GN, see [1, 2, 3]) constitute one of the well-developed alternatives of a methodology of describing processes and algorithms in formal and abstract way. The concept of generalized nets is an extension of the concept of Petri Nets, and the rest of their modifications and extensions.

2 TLS as an object of modelling

According to [5, 8], the following definitions will be needed.

Session negotiation on the control port: The server listens on the normal FTP control port and the session initiation is not secured at all. Once the client wishes to secure the session, the AUTH command is sent and the server *may* then allow TLS negotiation to take place.

Client wants a secured session: If a client wishes to attempt to secure a session, then in accordance, send the AUTH command with the parameter requesting TLS. The client then needs to behave according to its policies depending on the response received from the server and also the result of the TLS negotiation. A client that receives an AUTH rejection *may* choose to continue with the session unprotected if it so desires.

Server wants a secured session: The FTP protocol does not allow a server to directly dictate client behavior; however, the same effect can be achieved by refusing to accept certain FTP commands until the session is secured to a level that is acceptable to the server.

A scheme of establishing and work on secure FTP with TLS is presented in Figure 1.

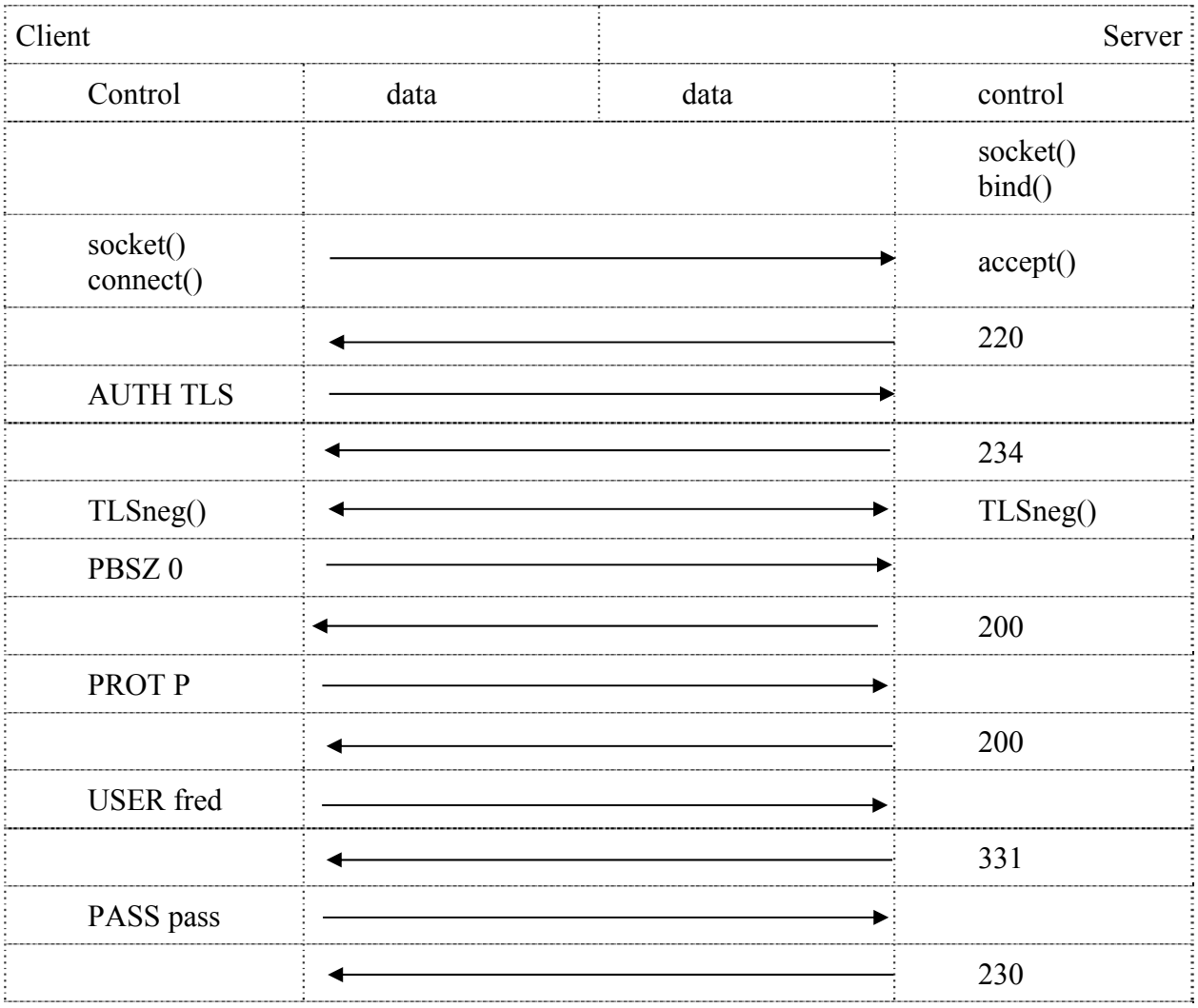

Figure 1. Establishing and work on secure FTP with TLS.

According to [9], Cyclic Redundancy Check (CRC) codes are shortened cyclic codes used for error detection. This technique, first of all a bit string called the generator polynomial is determined. It is called the generator polynomial because CRC is treated as a polynomial operation where input and generator bit strings are represented as polynomials with coefficients of 0s and 1s. A generator polynomial is used to generate checksums, which are appended at the end of frames. The receiver checks the input by using the same generator to detect transmission errors. There are two rules in selecting generator polynomials:

- They must be shorter than the frame length;
- They must start and end with 1.

If the text is attacked on its way to the receiver and some bits are changed, the corresponding bits in the decrypted plain text will also be changed. Also, if an attacker can capture two cipher texts encrypted using the same key stream and XOR these two cipher texts, then this result is also the XOR of the two appropriate plain texts. Having gained and collected this information, the attacker can use statistical attacks to recover the plain texts.

The more frequently a key stream is used for encryption and is captured by an attacker, the more easily he can perform statistical attacks. When the attacker recovers one of the plain texts, he can also recover the others.

The integrity check field referred to above is a checksum with 32-bit CRC and is also encrypted.

Each packet has CRC through which we excerpt information if it is whole or there it has some losses on it. Transmission over TCP give reliable delivery. Sequence numbers are used to coordinate which data has been transmitted and received. TCP will arrange for retransmission if it determines that data has been lost. We estimate μ and ν based on this information.

3 Intuitionistic fuzzy estimations of establishing secure File Transfer Protocol connections

Establishing of secure File Transfer Protocol connections is a process which depends on different factors, and is not always successful. This may depend on stability of the connection, closed connection by the client, the size and number of sent messages, syntax errors, denied request for policy reasons, etc. For this reason, it is worth making certain estimations of past connection rates which can be taken into consideration in the current time moment.

Our estimations of establishing secure File Transfer Protocol connections in each session are represented by ordered pairs ⟨μ, ν⟩ of numbers from the set [0; 1], where:

- μ is the number of sent and confirmed messages divided by the total number of sent messages. The value of μ is calculated as $\mu = \frac{C}{M}$, where
  - M is number of all sent messages in the current session;
  - C is number of sent and confirmed messages from the FTP server in the current session.
- ν is the number of sent and unconfirmed messages divided by the total number of sent messages. The value of ν is calculated as $\nu = \frac{U}{M}$, where

- U is number of sent and unconfirmed messages from the FTP client in the current session.

The degree of uncertainty $\pi = 1 - (\mu + \nu)$ reflects the number of sent and received messages, which have not been yet confirmed by recipient, divided by the total number.

4 Generalized net model

The generalized net model (for generalized nets see [1, 2, 3]), constructed in Figure 2 describes the process of establishing secure FTP with TLS, and its estimation in terms of intuitionistic fuzziness. In the beginning, the TLS starts a handshake and session protection fits into the existing logic of the FTP protocol. Initially, the following tokens are required to enter the generalized net:

- In position l_{11} a token enters with initial characteristic „*client socket, connect*";
- In position l_{2A} a token enters with initial characteristic "*server socket, bind*";
- In position l_{12}, l_{21}, l_{31}, l_{41} tokens have more than one characteristic.

It is developed a generalized net model with a set A of five transitions, having the form:

$$A = \{Z_1, Z_2, Z_3, Z_4, Z_5\},$$

where the transitions describe the following processes. Tasks performed by *FTP client* take place in transitions Z_1 and Z_3, while tasks performed by *FTP server* take place in transitions Z_2 and Z_4. Calculating of intuitionistic fuzzy estimations takes place in transition Z_5.

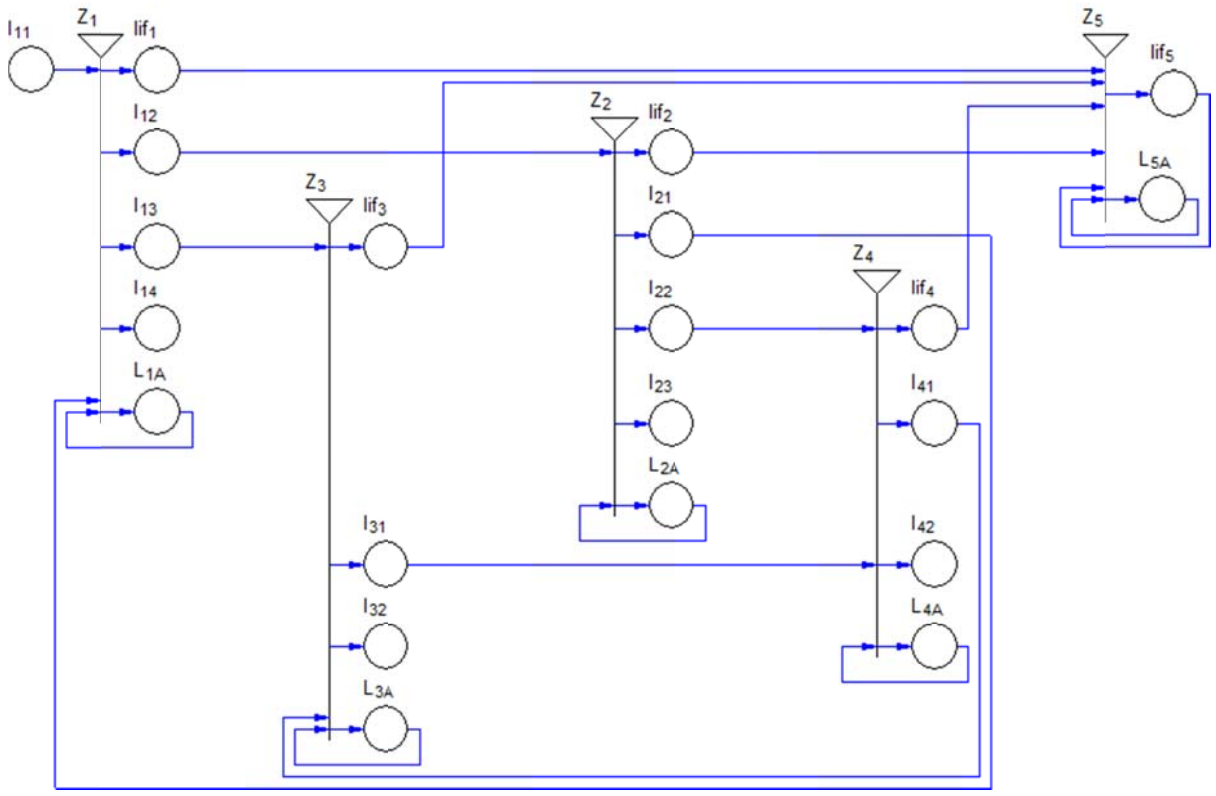

Figure 2. GN-Model of establishing secure FTP with TLS

Each transition Z_i has an index matrix of predicates, which define the conditions for transition of tokens from the transition Z_i's input to its output. These index matrices define the logic of the process, and are subsequently given below.

Transition Z_1 has the form:

$$Z_1 = \langle \{l_{11}, l_{21}, L_{1A}\}, \{l_{If1}, l_{12}, l_{13}, l_{14}, L_{1A}\}, R_1, \wedge (l_{11}) \vee (l_{21}, L_{1A}) \rangle$$

$$R_1 = \begin{array}{c|ccccc} & l_{If1} & l_{12} & l_{13} & l_{14} & L_{1A} \\ \hline l_{11} & false & W_{11,12} & false & false & true \\ l_{21} & false & false & W_{21,13} & false & true \\ L_{1A} & W_{1A,If1} & false & false, & W_{22,14} & true \end{array}$$

where:

- $W_{1A,If1}$ = "current token is sent from client";
- $W_{11,12}$ = "available are: *The FTP protocol specification dictates that it is up to the client to specify session attributes like the protection level. The server cannot require that the client use TLS, but it can refuse to accept any command from the client until it sends an AUTH TLS FTP command to upgrade the control channel to TLS protection*" and "available is: *234 AUTH command OK. Initializing SSL connection*";
- $W_{21,13}$ = "available is: *TLSneg from FTP server*";
- $W_{1A,14}$ = "available are: *421 Service not available, closing control connection*, *431 Need unavailable resource to process security, 500 Syntax error, command unrecognized*, *501 Syntax error in parameters or argument*, *502 Command not implemented*, *504 Command not implemented for that parameter, 534 Request denied for policy reasons.*

The token entering place l_{12} obtains new characteristics "sending data with: *client socket, connect*".

Transition Z_2 has the form:

$$Z_2 = \langle \{l_{12}, L_{2A}\}, \{l_{If2}, l_{21}, l_{22}, l_{23,} L_{2A}\}, R_2, \vee (l_{12}, L_{2A}) \rangle$$

$$R_2 = \begin{array}{c|ccccc} & l_{If2} & l_{21} & l_{22} & l_{23} & L_{2A} \\ \hline l_{12} & false & false & false & false & true \\ L_{2A} & W_{2A,If2} & W_{2A,21} & W_{2A,22} & W_{2A,23} & true \end{array}$$

where:

- $W_{2A,If2}$ = "current token is sent from server";
- $W_{2A,21}$ = "available are: *Common Responses 234 AUTH command OK. Initializing SSL connection*" and "available is: *TLSneg from ftp client*";
- $W_{2A,22}$ = "available is: *TLSneg*";
- $W_{2A,23} = W_{1A,14}$.

The token entering place l_{21} obtains new characteristics "available are: *client socket, connect, the 220 reply code is sent in response to a new user connecting to the FTP server to indicate that the server is ready for the new client*".

Transition Z_3 has the form:

$$Z_3 = \langle \{l_{13}, l_{41}, L_{3A}\}, \{l_{If3}, l_{31}, l_{32}, L_{3A}\}, R_3, \vee (l_{13}, l_{41}, L_{3A}) \rangle$$

$$R_3 = \begin{array}{c|cccc} & l_{If3} & l_{31} & l_{32} & L_{3A} \\ \hline l_{15} & false & false & false & true \\ l_{41} & false & false & false & true \\ L_{3A} & W_{3A,If3} & W_{3A,31} & W_{3A,32} & true \end{array}$$

where:

- $W_{3A,If3}$ = "secured current token is sent from client";
- $W_{3A,31}$ = "*Protection Buffer Size* (PBSZ) is sent" and "*Data Channel Protection Level* (PROT P) is sent" and "*Authentication username* (USER) is sent" and "*Authentication password* (PASS) is sent";
- $W_{3A,32} = W_{2A,23} = W_{1A,14}$.

Transition Z_4 has the form:

$$Z_4 = \langle \{l_{22}, l_{31}, L_{4A}\}, \{l_{If4}, l_{41}, l_{42,} L_{4A}\}, R_4, \vee (l_{22}, l_{31}, L_{4A}) \rangle$$

$$R_4 = \begin{array}{c|cccc} & l_{If4} & l_{41} & l_{42} & L_{4A} \\ \hline l_{22} & false & false & false & true \\ l_{31} & false & false & false & true \\ L_{4A} & W_{4A,If4} & W_{4A,41} & W_{4A,42} & true \end{array}$$

where:

- $W_{4A,If4}$ = "secured current token is sent from server";
- $W_{4A,41}$ = "*the 200 reply code is sent* (The requested action has been successfully completed)" and "*the 331 reply code is sent* (User name okay, need password)" and "*the 230 reply code is sent* (User logged in, proceed. Logged out if appropriate)";
- $W_{4A,42} = W_{3A,32} = W_{2A,23} = W_{1A,14}$.

Transition Z_5 has the form:

$$Z_5 = \langle \{l_{If1}, l_{If2}, l_{If3}, l_{If4}, l_{If5}, L_{5A}\}, \{l_{If5}, L_{5A}\}, R_5, \vee (l_{If1}, l_{If2}, l_{If3}, l_{If4}, l_{If5}, L_{5A}) \rangle$$

$$R_5 = \begin{array}{c|cc} & l_{If5} & L_{5A} \\ \hline l_{If1} & false & true \\ l_{If2} & false & true \\ l_{If3} & false & true \\ l_{If4} & false & true \\ l_{If5} & false & true \\ L_{5A} & W_{5A,If5} & true \end{array}$$

where $W_{5A,If5}$ = "results are estimated".

The token entering at place L_{5A} obtains characteristic "estimations $\langle \mu_i, \nu_i \rangle$". All estimations take initial values of $\langle 0, 0 \rangle$, when $i \geq 0$, the current $(i+1)^{-st}$ estimation is calculated on the basis of the previous estimations according to the iterative formula:

$$\langle \mu_{i+1}, \nu_{i+1} \rangle = \langle \frac{i.\mu_i + \mu}{i+1}, \frac{i.\nu_i + \nu}{i+1} \rangle,$$

where $\langle \mu_i, \nu_i \rangle$ are the previous estimations and $\langle \mu, \nu \rangle$ is the latest estimation of the messages for $\mu, \nu \in [0, 1]$ and $\mu + \nu \leq 1$.

In this way the token in place l_{if5} forms the final estimation of messages on the basis of the previous and the latest events.

5 Conclusion

The Generalized Net model described here is a possible model for the process of establishing secure FTP with TLS. The purpose of the present work is to describe a method of establishing secure FTP with TLS, using intuitionistic fuzzy estimations of the reliability of the connection. Applying of hierarchical operators, which could model the same transition at each place in more detail, would make the model more concrete.

The Transport Layer Security (TLS) support for File Transfer Protocol (FTP) is done in a similar way to the Simple Mail Transfer Protocol (SMTP) in [6] and Hypertext Transfer Protocol (HTTP) in [7]. The research done here for modelling the TLS protocol can be further applied to SMTP and HTTP, which are more widely used protocols, and face similar problems that call for the use of intuitionistic fuzzy estimations and modelling with generalized nets.

References

[1] Atanassov, K. *Generalized Nets*, World Scientific, Singapore, London, 1991.

[2] Atanassov, K. *On Generalized Nets Theory*. Prof. M. Drinov Academic Publ. House, Sofia, 2007.

[3] Atanassov, K. *On Intuitionistic Fuzzy Sets Theory*. Springer, Berlin, 2012.

[4] Atanassov, K., Generalized index matrices. *Compt. Rend. de l'Academie Bulgare des Sciences*, Vol. 40, 1987, No. 11, 15–18.

[5] Network Working Group, IBM UK Ltd, 2005, `http://tools.ietf.org/html/rfc4217`

[6] Network Working Group, Internet Mail Consortium, 2002, `http://www.ietf.org/rfc/rfc3207.txt`

[7] Network Working Group, Agranat Systems, Inc, 2000, `http://www.ietf.org/rfc/rfc2817.txt`

[8] Network Working Group, Bellcore, 1997, `http://tools.ietf.org/html/rfc2228`

[9] Network Working Group, Agilent, 2002, `http://tools.ietf.org/html/rfc3385`