The Secure Distributed Data Exchange Protocol (SDXP)

Erick Mabusi
Faculty of Information Technology
Strathmore University
Nairobi, Kenya
emabusi@gmail.com

Bernard Shibwabo Kasamani
Faculty of Information Technology
Strathmore University
Nairobi, Kenya
bshibwabo@strathmore.edu

Abstract*—*Distributed protocols implementations over a large network is a well-studied problem that converges asymptotically; however, existing protocols do not provide a way for each node to distributively detect the level of trust of another node. In this paper a method is developed to distributively determine whether a certain node should be trusted or not. In absence of such a method all nodes in the network keep communicating and running various computations even a certain node is known to be the origin of unwanted traffic, which is not preferable as in large-scale distributed networks resources like power are limited. Moreover, this additional communication can cause signal interference with other critical information. This distributed data security protocol is expected to take finite time and occurs at each node simultaneously.

Keywords—Data Networks; Network Security; Distributed Security; Nodes Security; Protocol

I. INTRODUCTION

Global organizations and large, multi-site institutions including universities, hospitals or government agencies are progressively facing a common challenge – how to bring all of their disparate locations into an easy to manage enterprise security system. On a distributed communications network, each station is connected to all adjacent stations rather than to a few switching points as in a centralized system [1]. In formulating an Enterprise security solution, security personnel in organizations focus on the following three priorities:

(i) Maintain a single storage of all personnel such that it need to be updated only once, across all locations or sites.

(ii) Deliver both central management and reporting, AND local site control

(iii) Protect the system and infrastructure against network failures

Organizations with different security software solutions at each site have success with the third priority, but may fail the first two. Formulating a true Enterprise security solution that is both easy to use and provides all the three priorities needs security software that can effectively manage the bulks of data generated by large Enterprises via Distributed Network Architecture. Fig. 1 presents a typical distributed network setup.

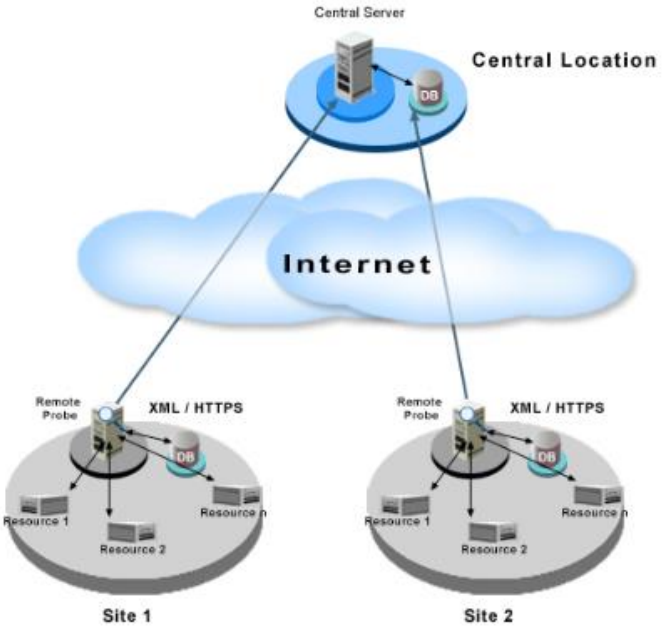

Fig. 4. A Typical distributed Network Setup

The main elements of Distributed Network Architecture are, the distribution of decision-making and control out to each site, while simultaneously, networking and synchronizing the various sites together via a central hub. Distribution of decision-making and control to each site is for two reasons, the first being for flexibility to manage the security needs specific to a site without being dependent on network connectivity and bandwidth back to a central, off-site server. The second reason, from a scalability perspective, well designed security architectures avoids unnecessary data transmission [2].

Distributed Network Architectures offer the local sites the information and capabilities to manage local security decisions autonomously, thus improving the scalability of the system and making it exceedingly tolerant of network failures and bandwidth shortages [2].

The benefits of Distributed Network Architecture are as follows [2]:

(i) Scalability:

Systems that depend on a single Enterprise server certainly suffer from performance issues as an organization grows and the server is overwhelmed. Furthermore, single server solutions are highly vulnerable to network failures.

(ii) Efficiency:

Security administrators control the flow of data and decision-making. Local data can be transmitted to each individual site and minimize the required network bandwidth.

(iii) Cost:

Hardware (Servers) and software at each local site can be suitably sized to meet the particular needs of each site, without necessitating installing an expensive server at even the smallest sites.

(iv) Reliability:

A Distributed Network Architecture is much more tolerant of network and hardware failures than a single server approach.

Centralized networks are usually vulnerable since the destruction of one node destroys the communications between the existing end stations [1]. In practice, a set of start and mesh components are used to form a communication network. In the case of type (b) in Fig. 2, a hierarchical network of a set of stars connected in the form of a larger star with an additional link forming a loop. In this case, the network is referred to as decentralized network since complete reliance upon a single point is not always necessary.

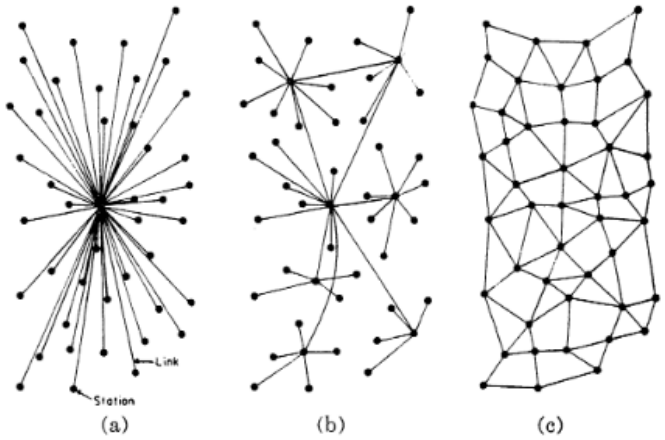

Fig. 5. (a) Centralized. (b) Decentralized. (c) Distributed networks.

Distributed networks offer redundancy levels that are difficult to achieve with centralized networks. Usually, a minimum span network formed with the smallest number of links possible, is selected as a reference point and is called “a network redundancy level 1”. In case two times as many links are used in a gridded network than in a minimum span network, the network is said to have a redundancy level of two. Fig. 3 presents some of the redundancy levels [1].

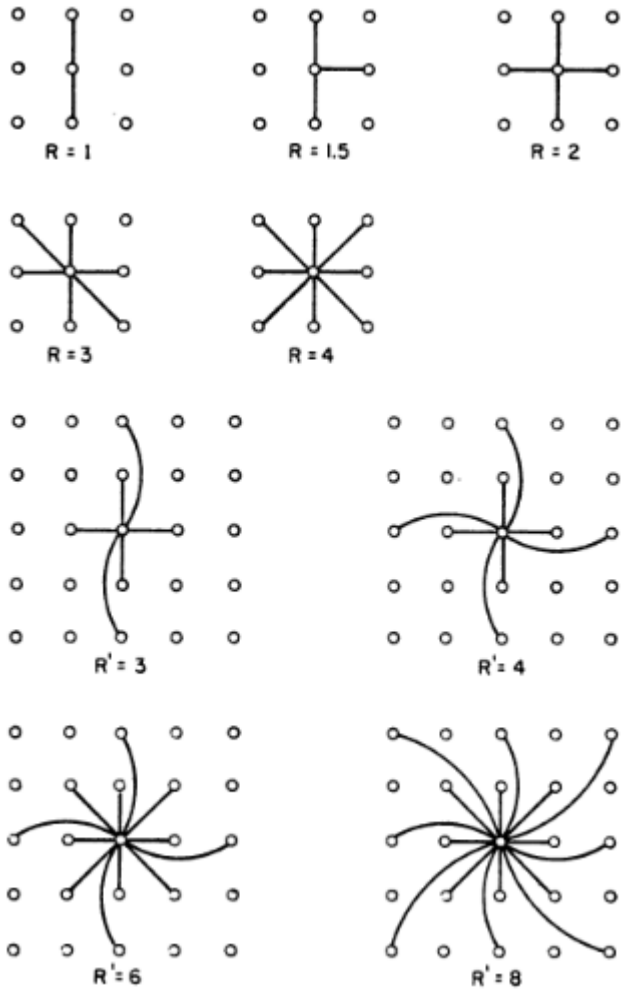

Fig. 6. Redundancy Levels in a Distributed Network

The TCP/IP distribution makes use of a handshake that expects a connection-based protocol. In this case, the protocol does not include any authentication after the handshake procedure. This is not completely safe, since it is vulnerable against takeover attacks. However, it is a tradeoff between fair safety and performance.

II. STANDARD DATA TRANSFER PROTOCOLS

A. FTP (File Transfer Protocol):

FTP is an application protocol that uses the Internet’s Transmission Control Protocol (TCP)/IP protocols. FTP is commonly used to transfer Web page files from their creator to the computer that acts as their server for everyone on the Internet. It’s also used to download programs and other files to your computer from other servers.

B. FTPs (File Transfer Protocol Secure — aka FTP over SSL):

FTPs is a protocol for transferring files using Secure Sockets Layer (SSL) to secure the commands and data that are being transferred between the client and the server.

C. SSH FTP (Secure Shell File Transfer Protocol — aka SFTP):

SSH FTP uses SSH to transfer files and requires that the client be authenticated by the server. Commands and data are encrypted to prevent passwords and other sensitive information from being exposed to the network in plain text.

D. HTTP (Hypertext Transfer Protocol):

The foundation of data communication for the Internet, this application protocol is the one to exchange or transfer hypertext. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers take in response to various commands.

E. HTTPs (Hypertext Transfer Protocol Secure — aka HTTP over SSL):

HTTPs is a secure version of HTTP and it allows secure e-commerce transactions. Using HTTPs, computers agree on a code between them on a Secure Sockets Layer, and then they scramble the messages using that code so that no one in between can read them.

F. MLLP (Minimal Lower Layer Protocol):

Commonly used within the HL7 (Health Level Seven) community for transferring HL7 messages, MLLP provides a minimalistic session-layer framing protocol. MLLP supports only direct connections between a sender and a receiver, and there is no authentication process.

G. AS1 (Applicability Statement 1):

AS1 is a data transfer standard that used the email protocol to move data and is largely unused in practice today for systematic file exchange. Disadvantages of AS1 include congregating AS1 payloads with regular email, and a delivery mechanism subject to vagaries of email relays, latencies, and loss of message control. Due to the low market adoption of AS1, it is not covered further in this paper.

H. AS2 (Applicability Statement 2):

AS2 is a standard by which users transfer EDI or other data, such as Extensible Markup Language (XML) or plain text documents, over the Internet using HTTP and HTTPs. AS2 offers increased verification and security achieved through the use of receipts, digital signatures, and file encryption. Its transactions and acknowledgments occur in real time, increasing the efficiency of document exchanges.

I. AS3 (Applicability Statement 3):

AS3 is the IEFT messaging specification standard that enables software applications to systematically communicate data, including EDI and XML, over the Internet using file transfer protocol (FTP). AS3 is not the next version of AS2 as it offers its own unique features and provides security for the transport payload through digital signatures and data encryption.

J. AS4 (Applicability Statement 4):

AS4 provides guidance for a standardized methodology for the secure and document-agnostic exchange of B2B payloads using Web Services. The profile focuses on providing an entry-level onramp for Web Services B2B messaging. ebMS 2.0 (ebXML Messaging Service)

K. SMTP (Simple Mail Transfer Protocol):

This a protocol for sending email messages between servers. Most e-mail systems that send mail over the Internet use SMTP to send messages from one server to another; the messages can then be retrieved with an e-mail client.

L. SMTPs (Simple Mail Transfer Protocol Secure — aka SMTP over SSL):

This is a method for securing SMTP with transport layer security. It is intended to provide authentication of the communication partners. SMTPs is not an extension of SMTP; it is just a way to secure SMTP at the transport layer

III. PROBLEM DEFINITION

Exchanging information across digital networks requires each party to speak the same data transport language. A single organization may likely speak a variety of languages. In information technology, protocols enable file transfers by outlining a standard procedure for regulating the data exchange between businesses. Protocols specify interactions between the communicating entities, and they can often be prescribed by industry or other standards.

Managed file transfer protocol flexibility becomes increasingly important as a network grows. At the core of evolving protocols is usually the expectation for heightened security around the data being transferred. The functional elements of protocol security include privacy, authentication, integrity, and non-repudiation [3, 4].

Some of the major transfer needs for an organization include file size, volume, frequency, certificate management, and the associated functionality required. Depending on the needs and trading partner requirements, one or many protocols may be appropriate for an organization.

IV. DEFINITIONS

For purposes of understanding of the proposed protocol in detail the following introductory definitions are essential to convey their importance in the protocol. See Table 1 for the definitions.

TABLE I. PROTOCOL DEFINITIONS

TERM	DEFINITION
Node	A node is any entity on the distributed network that needs to implement a form of security or authentication in it. Nodes communicating by using the SDXP

	protocol participate in exchange of data with the network or other nodes in the network. Nodes store a ledger of data for the protocol. Example of nodes types can include • Desktop operating systems • Web based application • Desktop application • Content Delivery Networks • Internet Service Provider • Virtual Networks • Mobile operating systems • IoT based devices • Physical security company • Gaming devices
Environment	Intrinsic information of about a node that distinguishes it from another node or type of node for example BIOS version and OS type/version. This can help to filter security concerns related to a specific or a group of environment variables
Environment variable	A single data that makes an environment for example the OS type.
Accessor	An accessor is any entity that is performing a certain action on a node. Examples of accessors • Another SDXP node • A user on a computer • A user of an application program • A device from network • A program • A mobile phone user • IoT devices • Bots
Action	An action is the process of interacting or performing an act on a node with respect to security. Actions are performed by an accessor and data related to actions and/or its accessor are exchanged (shared) via the protocol.

	There can be 4 main types of actions which are: • Good • Bad • Neutral • Attempt In a group or public data exchange the node environment is also embedded to the action.
Score	An Action score is the weight in terms of severity of a specific action or action type. This can based on the nature of business of the node or impact that can be caused by the action or action type. There are 3 types scores • Reserved: These are scores for known actions by the protocol itself example address ping • Public: These are public scores for the actions • Private: These scores are only private to selected nodes and take precedence over the other scores. There are two categories of private scores: o Single – this originates from the node itself. o Group - this originates from the group within which the node belongs.
Condition	Single piece of information about an action that are NOT related to an accessor
Note	Single piece of information about an accessor collection during an action or manually by the node taking example an IP address and name respectively
Log	Any piece of information stored for security concern. Example a note or a condition
Statement	A chain collection of all logs about an accessor or action.
Ledger	Collection of statement(s) stored by a node for the purpose of making decisions on security concerns.
Registry	The database of containing the ledger and other protocol based data

V. GENERAL IDEA

Exchanging information across digital networks requires each party to speak the same data transport language. The purpose of the protocol is to allow nodes to share any data relating to the security of a node with a private group or a

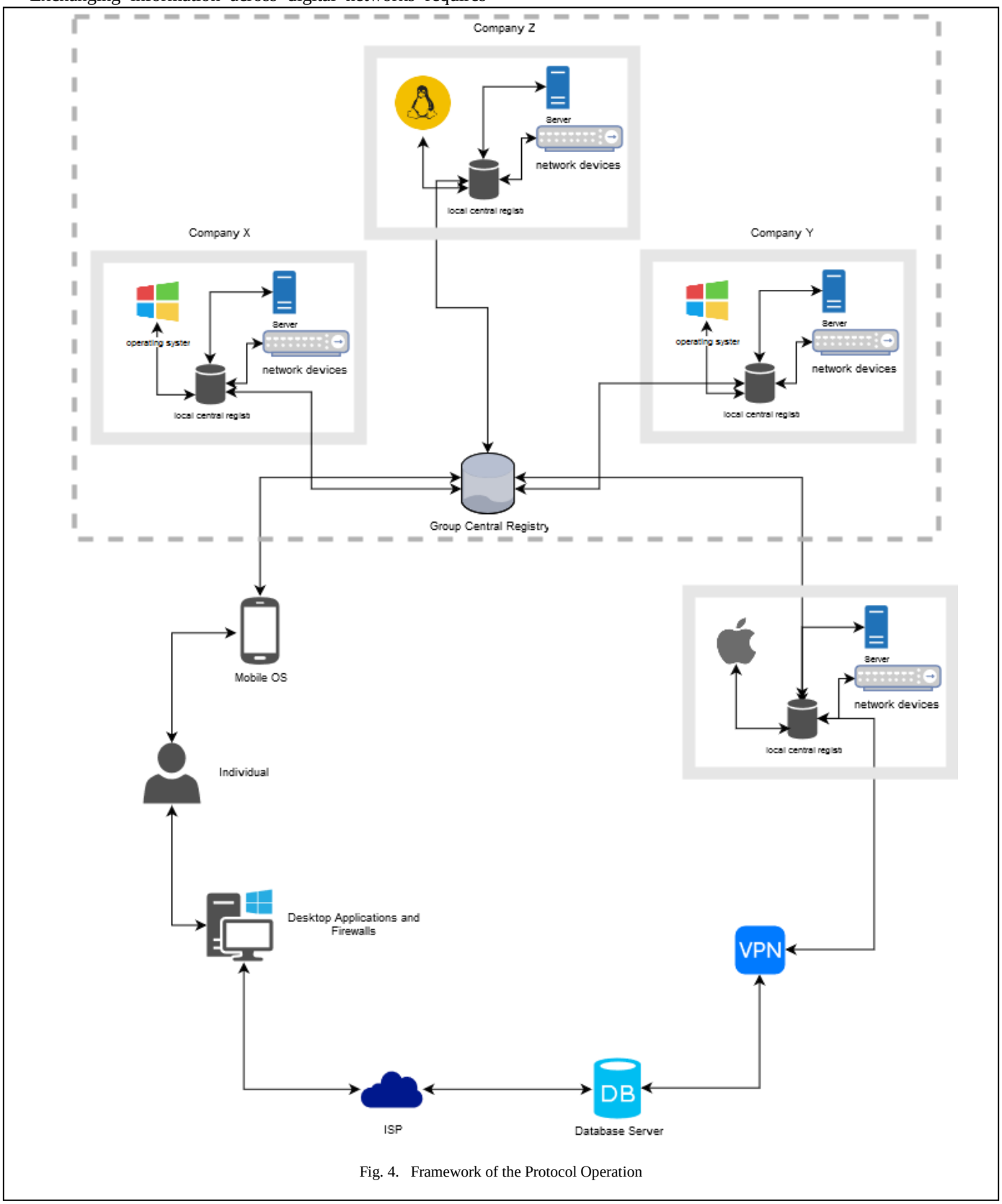

Fig. 4. Framework of the Protocol Operation

public one. The shared data between nodes can then be further interpreted or used to address security concerns, for example, early threat detection at the node level.

An initial node setup of the protocol establishes the following basic configurations and resources:

- A unique identifier for the node during exchange of data
- The node's environment variables including their specific share preference.
- A private key and public key to be used by the node for encryption of exchange data.
- A local registry containing a ledger of all logs and protocol based data.
- The exchange type preference for the node, this is either to enable the protocol to automatically monitor and record logs to the registry or manually recording logs prompted by node (or node component) by issuing command.
- The data exchange preference for the node, this is the sharing preference of the local registry on participating in a group or public security data exchange
- NOTE: A node can have components that can be termed as a node, during initial setup such can be optionally setup simultaneously other that just being environment variables.

Each node in the protocol has its registry locally which stores logs of all components of the node for example a machine hosting an application X can be setup such that the business application, operating system and any devices connected to it all register security issues to the protocol.

In a group or public data exchange each node stores also its own local data while allowing other nodes to access its data when querying. Group policies can be set to have some nodes be a central registry for all the group to allow faster querying and act as back up to retrieve data when a node is offline. The public protocol by default has nodes that store all data exchanged in order to ensure no public data is lost. In essence the public exchange is a special group exchange that involves all node in the protocol regardless of their node type.

Data is exchanged from a node immediately after storage in the local registry, this ledger data embedded with the node's environment variables are hashed using public key cryptography to be received by the requesting node or stored in a group central node. When a certain log type already exists, a data chain is added to the entry as log vote with any specific additional details.

A node can query data from a specific node in group exchange or from the whole group by querying the group registry. A query can be for general data or for specific accessor, condition, note or environment variable, such queries are returned in chain form and includes the logs vote. Fig. 4 shows the framework of the protocol operation.

VI. COMMUNICATION

A. Types

The Node using the protocol can interact with the protocol using the following categories of communications

- Initialize
  This is an action in the communication that involves setting up the node and/or the node environment in the protocol. These include changing of configuration, share preference, basic details including changing key.
- Update
  This is a communication aspect that involves a node sharing ledger data in a group or to the public.
- Retrieve
  This is a communication aspect that involves a node retrieving data from its local, group or public registry.
- Stop
  This a special communication that stops the use of the protocol and/or sharing data to group or public registry.
- Special
  This is communication used internally by in the operation of protocol for example acknowledge message. Special communication also includes communication that allow node to perform action in the protocol like submit to be a group central node.

B. Data Exchange Formatting

The protocol allows support all standard data exchange formats in group communication as stated in (http://ieeexplore.ieee.org/document/579126/). For the public data exchange XML and JSON are the main data exchange format.

C. Encryption

The transmission of data in the protocol is encrypted using AES-256 encryption algorithm and passed through secure socket layer.

D. Encryption

Public key cryptography using a combination of public and private key is used to authenticate communication is sent from the authorized node.

VII. VOTING MECHANISM

Voting for ledger entry is done by other nodes by chaining data to log entries in the log and on the interpretation an average weight can be determined, a node in turn gets voting score as it votes too.

VIII. REGISTRY

The registry contains the following kinds of information

- Ledger
- Protocol based data

A ledger contains the following information

- The node accessors
- The node accessors note
- The node actions
- The node action's conditions

The protocol based data include the following

- Node identification details
- Node environment variables
- Node keys
- Node activity logs

In group based protocol

- Participating nodes
- Each node voting score

IX. SIMILAR APPLICATION AREAS OF THE PROTOCOL

The following are other areas where the methodology used in SDXP implementation can be applied in:

- Discovering the relation between nodes in security threat or attack
- National security, for example, tracking of criminal and terrorist activities and sharing data between governments
- Digital currency security enhancement

X. CONCLUSIONS

Protocols specify interactions between the communicating entities, and they can often be prescribed by industry or other standards. There are various protocols that are used in distributed computing including Connectivity test protocols, Minimum-hop-path protocols and Path-updating protocols. In this paper, we introduce a security based protocol for securing communication over a distributed networks. Various factors are described and defined under the context of a secure distributed network and application areas defined. With this protocol, the multiple autonomous nodes at each site can be in communication with the central hub via either a Local Area Network (LAN), Wide Area Network (WAN) or the public Internet securely.

XI. REFERENCES

[1] P. Baran, "On Distributed Communications Networks," IEEE Transl. of the Technical Group on Communications Systems, vol. CS-12, Number 1, March 1964.

[2] W. Brown, "Distributed Network Architecture: Scalability and Load Balancing in a Security Environment" Tyco Security Produsts.

[3] Cleo, "Secure Data Exchange Protocols: Comparing transport protocols to securely exchange business data within and beyond the enterprise", 2016

[4] A. D. Thurston. "DSNP: A Protocol for Personal Identity and Communication on the Web"