In Media Streaming Security –FocusData leakage and Modifiers in Mobile Environment

V. Saravanan[1], Dr.C.Chandrasekar[2]

[1]Asst.Professor in Computer Applications,HindusthanCollege of arts and Science, Coimbatore, India

[2]Associate Professor, Dept.of Computer Science, Periyar University, Salem, India

ABSTRACT

Generally, in theemblematic deployment of mobile networks, where a huge number of nodes are aimlessly deployed ina two dimensional area. Packet dropping and modification are frequent attacks that can be launched by an adversary to disrupt communication in even based mobile networks. Several schemes have been wished-for to take the edge off or put up with such attacks but very few can effectively and efficiently identify the intruders. To address this problem and make a security without affecting an even based streaming process, we propose a simple howevervaluablemethod, which can classifymischievous forwarders that drop or modify packets. Extensive analysis and simulations have been conducted to verify the effectiveness and efficiency of the scheme. Most of the bad nodes can be identified by our heuristic ranking algorithms with littleartificialencouraging.

KEYWORD

Even based Streaming, Packet Dropping, Intrusion Detection, Packet Modification, Even based Streaming, Mobile Networks

1. INTRODUCTION

In Live media streaming, for data transmit, we using huge number of nodes are aimlessly deployed in even based approach. In this Streaming approach we use both CBR and VBR, Compare to CBR, VBR gives a good result for using even based live media streaming. Security is a big issue for an even based approach. The exploitation of this processing model in the presence of XML security is not. Due to the lack of algorithms for event-based processing of Streaming-Security, most SOAP frameworks include the option for streaming based processing only for unsecured SOAP messages. As soon as these messages are secured by Streaming-Security mechanisms, the security processing is performed relying on the DOM tree of the secured message, hence loosingof all advantages in the event-based processing model. With this contribution, the implementation of SOAP message processing can be realized in a streaming manner including the processing of security means, resulting in significant improvements compared to “traditional” and currently mainly deployed tree-based approaches. The main advantages of the streaming model include an increased efficiency in terms of resource-consumption and an enhanced robustness against different kinds of Denial-of-Service Robustness(DoS) attacks. In existing system a Mobile network in even based approach is often deployed in an unattended and hostile environment to perform the monitoring and data collection tasks. When it is deployed in such an environment, it lacks physical protection and is subject to node compromise. After compromising one or multiple nodes, an adversary may launch various attacks to disrupt the in-network communication. Among these attacks, two common ones are packet leakage and packets modifying, i.e., compromised nodes drop or modify the packets that they are supposed to forward. In this research proposal, a trouble-free yet effective scheme to identify misbehaving forwarders that drop or mutate packets. Each packet is encrypted and padded with key so as to hide the source of the packet. The packet mark, a small number of extra bits, is added in each packet and key such that the sink can recover the source of the packet and then figure out the dropping ratio associated with every sensor node. The routing tree structure dynamically changes in each round so that behaviours of sensor nodes can be observed in a large variety of scenarios. Finally, most of the bad nodes can be identified by our heuristic ranking algorithms with small false positive. And then the streaming security will be added without affecting even based process, our proposed process done it in an apt move towards. This paper initiates the concepts and algorithms for a comprehensive stream-based Streaming-Security component. By implementing configurable chains of stream processing components, a streaming Security validator has been developed, which verifies and decrypts messages with signed and/or encrypted parts against a security policy. The solution can handle any order, number and nesting degree of signature and encryption operations filling the gap in the stream-based processing chain towards more efficient.

2. STIMULUS AND ENVIRONMENT

To set the scenes for this paper, some rousing foundations and related state-of-the-art are briefly introduced in the ensuing sections.

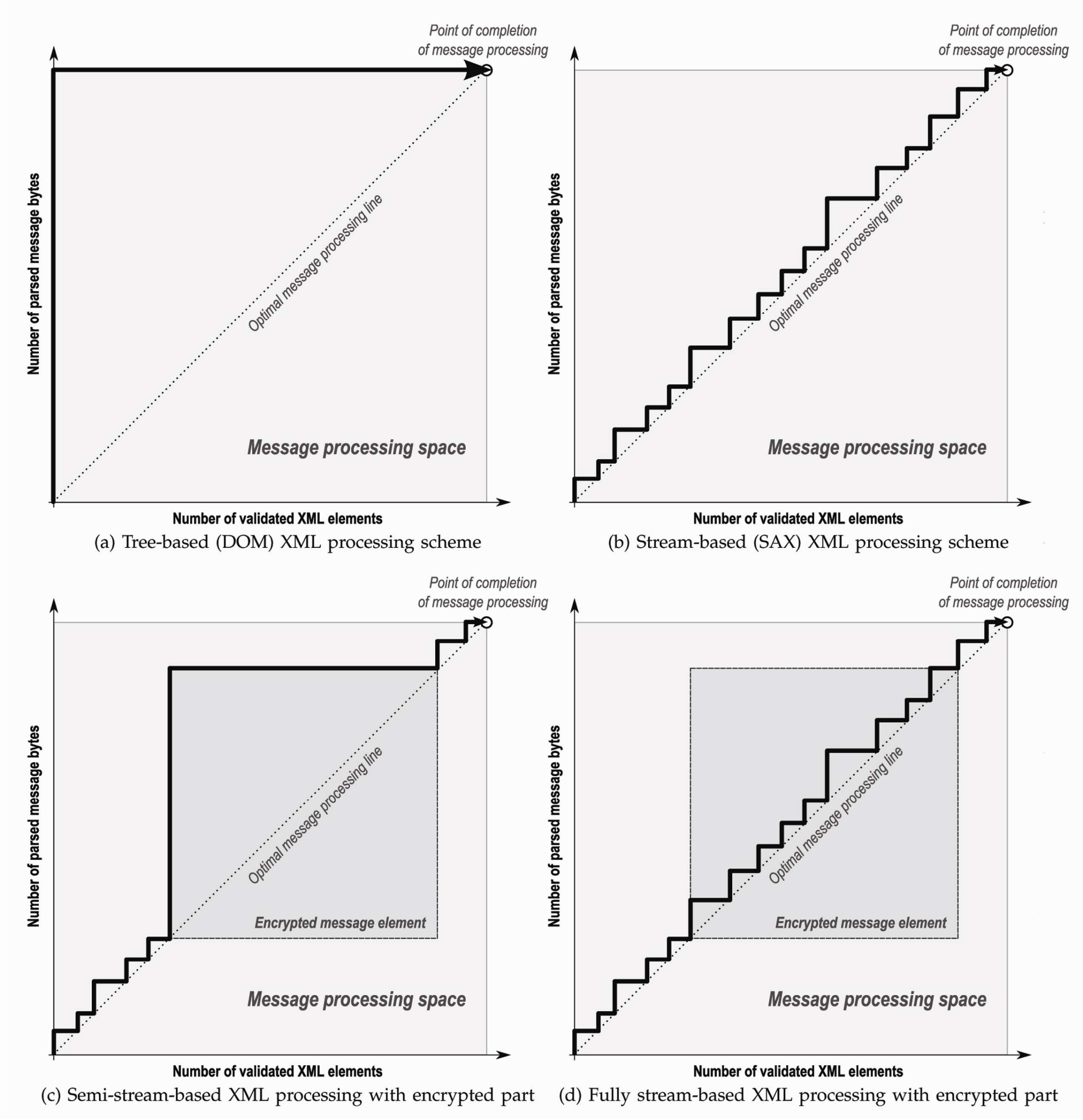
2.1 Network Supposals

We consider a typical deployment of sensor networks, where a large number of sensor nodes are randomly deployed in a two dimensional area. Each sensor node generates sensory data periodically and all these nodes collaborate to forward packets containing the data towards a sink. The sink is located within the network. We assume all sensor nodes and the sink are loosely time synchronized [15], which is required by many applications.Attack-resilient time synchronization schemes, which have been widely investigated in wireless sensor networks [16], [17], can be employed. The sink is aware of the network topology, which can be achieved by requiring nodes to report their neighboring nodes right after deployment.

When applying the second model, the XML document is read and parsed step by step, divided into parts (e.g. single XML elements) and passed to the application. The application then operates on these XML parts. One can distinguish two approaches for implementing the communication between the parser and the application: a pull and a push approach. With the pull approach, the application requests the next XML part from the parser. A well-known implementation of this approach is the Streaming API for XML. Using the push approach, the parser calls the application, which waits for incoming XML events. That's why this model is also called event-based XML processing. The most common realization of the event-based model is the Simple API for XML (SAX). Streaming processing has generally a large advantage regarding resource consumption. As no object tree representing the complete XML document has to be created, especially the memory consumption for streaming processing is much lower than for document-based processing.

2.2 Slipup Finding

Assuming the parsed XML document to contain an XML Schema violation (while still being well-formed), the flaws of the tree-based approach become clear. As shown in Figure 1 (a), the XML document must be read completely before any processing on the contained XML elements can start.This enablesa malicious SOAP message sender to feed in a hugeXML document of arbitrary, schema-violating contents,which the parser must read in completely before being able to detect the presence of a schema violation. Thisway, as the size of the SOAP message is in control ofthe sender, an attacker can cause heavy workloads forparsing XML documents. Results inan irregular stream of events for every XML element theparser extracts from the character blocks, as shown inFigure 1 (b).For the special case of an XML Schema violationwithin the processed document, the stream-based approachenables an early processing of the XML contents.Again, the more convenient approach consists in storingthe encrypted block completely in memory, then applyingthe cryptographic operation of decryption to it resulting in a new, plain XML fragment. Then, the newXML fragment is parsed and processed subsequently.This approach is shown in Figure 1 (c). A more efficient approach consists in decryptingencrypted XML document parts "on the fly", and havingtheir XML contents parsed and processed immediately. (Figure 1 (d)).


Point of completion of message processing
Number of parsed message bytes
Optimal message processing line
Message processing space
Number of validated XML elements
Encrypted message element

(a) Tree-based (DOM) XML processing scheme

(b) Stream-based (SAX) XML processing scheme

(c) Semi-stream-based XML processing with encrypted part

(d) Fully stream-based XML processing with encrypted part

Fig. 1: Graphical evaluation of different XML dealing out schemes

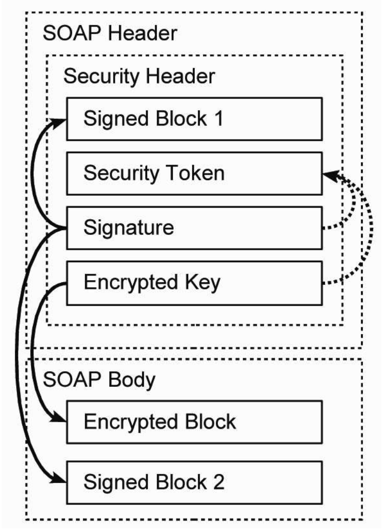


Fig.2: Streaming-Security protected SOAP communication

3.STREAMINGSECURITY PROCESSING

In this section, the algorithms for processing WS-Security enriched SOAP messages in a streaming manner are presented and discussed. To understand the algorithms and the problems solved by them, first of all an introduction of the WS-Security elements is given. Figure 2 shows an example of a Security-securedSOAP message containing these references. Security tokens contain identity information and cryptographicmaterialand are used inside signatures and encrypted keys and are backward referenced from those. Encrypted key elements contain a(symmetric) cryptographic key, which is asymmetrically encrypted using the public key of the recipient. This symmetric key is used for encrypting message parts (at the client side) and also for decrypting the encrypted blocks (at the server-side). Encrypted keys must occur inside the message before the corresponding encryptedblocks.Event-based processing is typically described using FiniteState Machineswith automation transitions triggered by the events.

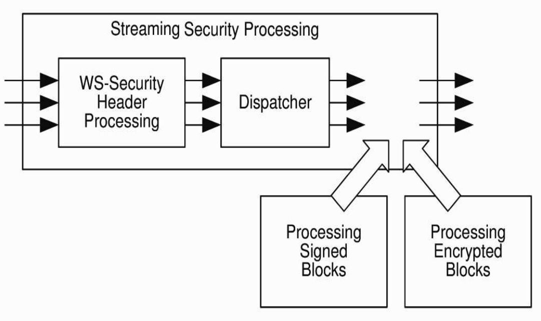

Streaming Security Processing
WS-Security Header Processing
Dispatcher
Processing Signed Blocks
Processing Encrypted Blocks

Fig. 3: structural design of a Stream-based Security dealing out System

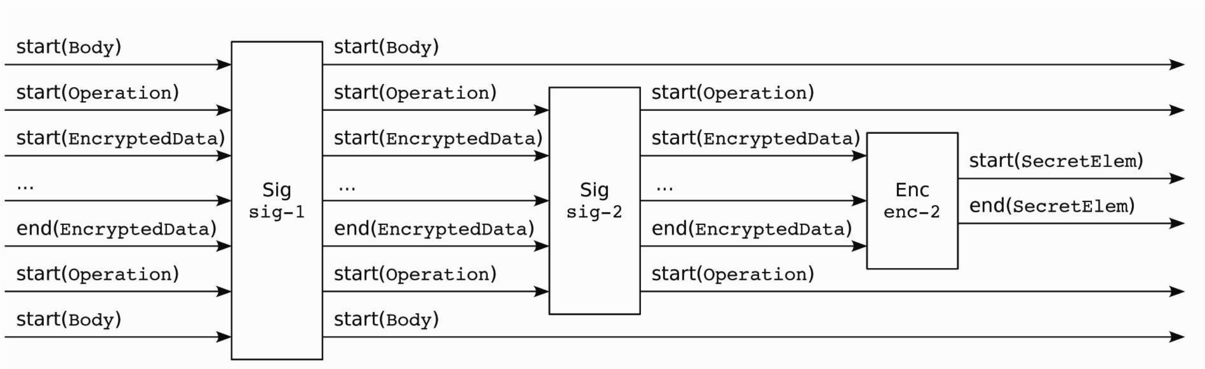


Fig. 4: Streaming -occurrencedispensation of Signed and Encrypted Blocks – Example 1 (Encrypt ahead of Signing)

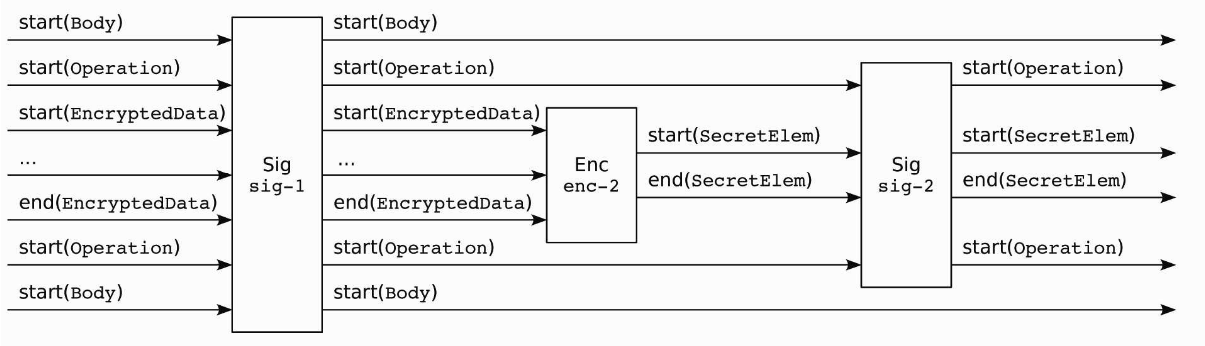


Fig. 5: Streaming -occurrencedispensation of Signed and Encrypted Blocks – Example 2 (Sign ahead of Encrypting)

To deal with packet droppers, a widely adopted countermeasure is multi-path forwarding [2], [3], in which each packet is forwarded along multiple redundant paths and hence packet dropping in some but not all of these paths can be tolerated. To deal with packet modifiers, most of existing countermeasures [6], [7] aim to filter modified messages en-route within a certain number of hops. These countermeasures can tolerate or mitigate the packet dropping and modification attacks, but the intruders are still there and can continue attacking the network without being caught. To locate and identify packet droppers and modifiers, it has been proposed that nodes continuously monitor the forwarding behaviours of their neighbours [12], to determine if their neighbours are misbehaving, and the approach can be extended by using the reputation-based mechanisms to allow nodes to infer whether a non-neighbour node is trustable. This methodology may be subject to high energy cost incurred by the promiscuous operating mode of wireless interface; moreover, the reputation mechanisms have to be exercised with cautions to avoid or mitigate bad mouth attacks and others.

4. VALUATION

The algorithms introduced in this paper provides a comprehensive framework for event-based mobile-Security processing.With this contribution, the implementation of SOAP message processing can be realized in a streamingmanner including the processing of security means,resulting in a significant increase in efficiency compared to“traditional” and currently mainly deployed tree-basedapproaches. First of all, if the SOAP message is invalidthe message is only read and parsed up to the pointwhere any flaw is detected. This is especially important,when the message was part of a Denial-of-Service attack,or if a negative Access Control decision can be made at an early stage of the message processing. Further,even if only valid messages are considered the eventbasedapproach has many advantages, especially withregard to memory consumption. It is well known that inmemorydocument trees are 10 to 100 times larger thanthe serialized form, while the XML events only increasethe size by factor 1 to 1.5.To prove these theoretical advantages, the approach presented here was prototypically implemented in a system calledCheck Way. This system was extensivelytested and compared to standard frameworks. The tests were performed on an Athlon 64 2500+ with 1 GB memory. The Java VM used is version 1.6.23 from Sun Microsystems.

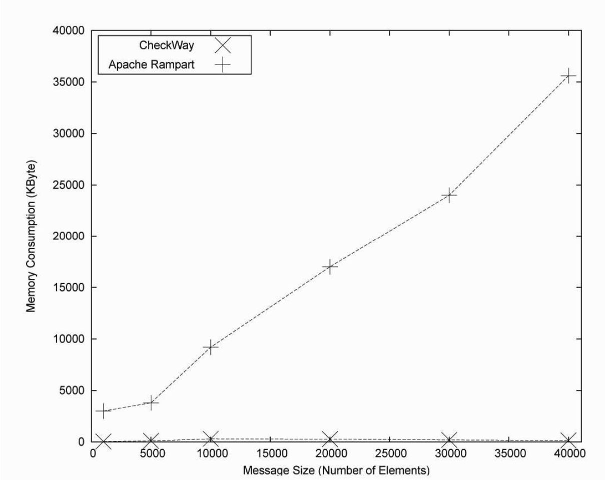


Fig. 6: Comparison of Memory Consumption

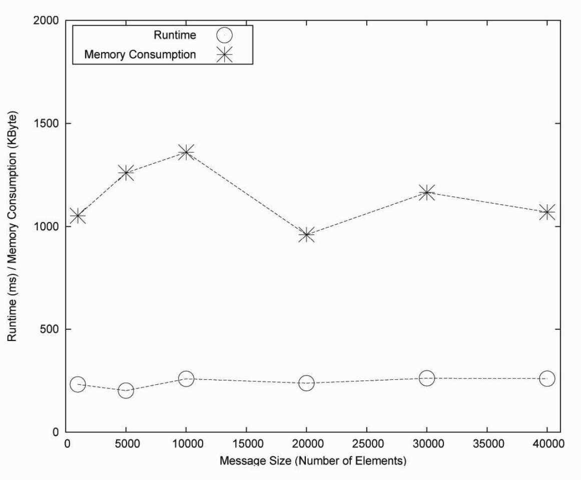


Fig. 7: Negative Access Control Decision

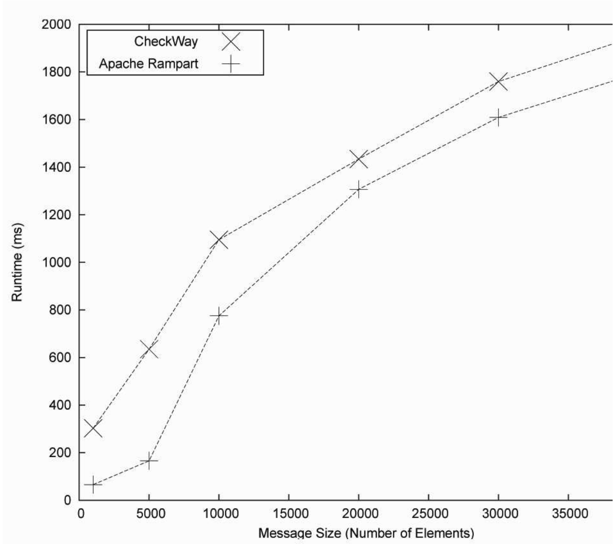


Fig. 8: Comparison of Runtime

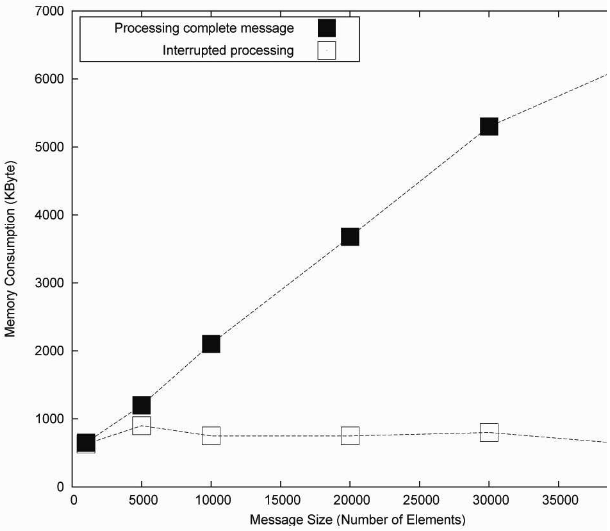


Fig. 9: Memory Consumption for complete and for early-interrupted processing

For the first solitary, Streaming invoked with aseries of messages with increasing number of elements inside the encrypted SOAP body. The test compared the resource consumption of CheckWay with Apache Rampart. Figures 6 and 7 shows the memory consumption and the runtime of both systems for differentmessagesizes. It is noticeable that the runtime for CheckWay is only slightly higher than for Rampart (convergingto approximately 10%). However, for the memory consumption Check Way is magnitudes better than Rampart.Rampart needs between 20 to 110 times more memory than our approach.

The second scenario illustrates that adoption of thestreaming approach not only leads to quantitative performanceimprovements but also has fundamental advantages compared to document-based tree models, the lower curve in Figures 8 shows the resulting memory consumption of the overall SOAP processing system One can see that the message is only processed up to the point where the policy violation occurs. In the third scenario, an access control violation is assumed. This occurs for example if an attacker has no valid authentication credentials for the service he isattacking.At the end of the security headera negative access control decision can be made, even ifthe attacker uses the credentials of a valid eavesdropped message. Using streaming processing of the SOAP message, the access control decision can be made at anearly stage of the security processing leading to constant runtime and memory consumption independent of the actual message size(Figures 9).

4.1 Subsequent Representation

And we proposed scheme consists of a system initialization phase and several equal-duration rounds of intruderidentificationphases.In the initialization phase, sensor nodes form a topology which is a directed acyclic graph (DAG). A routing tree is extracted from the DAG. Data reports follow the routing treestructure. In each round, data is transferred through the routing tree to the sink. Each packet sender/forwarder adds asmall number of extra bits to the packet and also encrypts the packet.

When one round finishes, based on the extra bits carried in the received packets, the sink runs a node categorizationalgorithm to identify nodes that must bebad and nodes that are suspiciously bad. The routing tree is reshaped every round. As a certain number of rounds have passed, the sink will have collected information about node behaviors in different routing topologies. The information includes which nodes are bad for sure, which nodes are suspiciously bad, and the nodes' topological relationship. To further identify badnodes from the potentially large number of suspiciously bad nodes and we change the path.

5. CONCLUSION

We propose a simple and efficient scheme to make outunrulyforwarders that plummet or modify packets. And each packet is encrypted and padded y Streaming Security. So, as to hide the source of the packet.The packet mark, a small number of extra bits, is added in each packet such that the sink can recover the source of the packet and then figure out the dropping ratio associated with everynode.The routing path structure dynamically changes in eachrounds so that behaviors of sensor nodes can be observed in a large variety of scenarios. Finally, most of the bad nodes can be identified by our heuristic ranking algorithms with smallartificial constructive. Extensive analysis, simulations and implementation have been conducted and verified the effectiveness of the proposed plan and another related research directionis heading towards thedevelopment of consistent security policies in large workflows.

6. ACKNOWLEDGMENTS

Our sincere thanks to the experts who have contributed towards the development of the research work.

7. REFERENCES

[1] H. Chan and A. Perrig, “Security and privacy in sensor networks,” Computer, vol. 36, no. 10, 2003.

[2] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” in the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.

[3] V. Bhuse, A. Gupta, and L. Lilien, “Dpdsn: Detection of packet-dropping attacks for wireless sensor networks,” in the Fourth Trusted Internet Workshop, 2005.

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y.Ahn, andS. Moon, “I Tube, You Tube, Everybody Tubes: Analyzing theWorld’s Largest User Generated Content Video System,” in *Proceedings of the ACM SIGCOMMConference on Internet measurement (IMC ’07)*, SanDiego, CA, USA, Oct 2007.

[5] B. Ager, F. Schneider, J. Kim, and A. Feldman,“Revisiting Cache ability in Times of User Generated Content,” in p*roceedings of 13th IEEE Global Internet Symposium 2010*, San Diego, CA, USA, Mar 2010.

[6] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Duratek. Scott, K. Fall, and H. Weiss, “Delay-Tolerant Networking Architecture,” IETF RFC 4838, Apr 2007.

[7] M. F. Arlitt and C. L. Williamson, “Internet Web Servers: Workload Characterization and Performance Implications,” *IEEE/ACM Transactions on Networking*,vol. 5, no. 5, pp. 631–645, 1997.

[8] Y. Zhu, M. Chen, and A. Nakao, “CONIC: Content-Oriented Network with Indexed Caching,” in *Proceedings of 13th IEEE Global Internet Symposium 2010*,San Diego, CA, USA, Mar 2010.

[9] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, “An acknowledgment-based approach for the detection of routing misbehavior inmantes,” *Mobile Computing, IEEE Transactions on*, vol. 6, no. 5,2007.

[10] J.W. S. P. Team, “Streaming APIs for XML parsers,” Sun Microsystems, Tech. Rep., 2005.

[11] The SAX Project, “Simple API for XML – sax 2.0.1,” http://www. saxproject.org/, 2002.

[12] R. Roman, J. Zhou, and J. Lopez, “Applying intrusion detection systems to wireless sensor networks,” in IEEE CCNC, 2006.

[13] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” ACM Transactions on Sensor Networks (TOSN), vol. 4, no. 3, 2008.

[14] W. Li, A. Joshi, and T. Finin, “Coping with node misbehaviors in ad hoc networks: A multi-dimensional trust management approach,” in IEEE Mobile Data Management, 2010.

[15] Q. Li and D. Rus, “Global clock synchronization in sensor networks,” in IEEE INFOCOM, 2004.

[16] K. Sun, P. Ning, C. Wang, A. Liu, and Y. Zhou, “Tinysersync: Secure and resilient time synchronization in wireless sensor networks,” in ACM CCS, 2006.

[17] H. Song, S. Zhu, and G. Cao, “Attack-resilient time synchronization forwireless sensor networks,” Ad Hoc Networks, vol. 5, no. 1, 2007.

V. Saravanan received M.Sc(CS)., from Bharathidasan university in 1999, completed M.Phil., from Manonmaniam Sundaranar university in 2002. Received MCA., from Periyar university in 2011. Currently working in Asst.Professor Dept. of Computer Applications, Hindusthan College of arts and science, Coimbatore. His research area is Network security in mobile networking.

Dr.C.Chandrasekar is an Associate Professor in Dept. of Computer Science, Periyar University, Salem. He has done MCA., Ph.D., He is been in the Teaching and Research field for more than 15 years. His research area is Mobile and Wireless computing. He has worked and published more than 40 research articles.