EVALUATION AND COMPARISON OF MULTICAST MESSAGE AUTHENTICATION PROTOCOLS FOR USE IN POWER GRID APPLICATIONS

By

THANIGAINATHAN MANIVANNAN

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER ENGINEERING

WASHINGTON STATE UNIVERSITY

School of Electrical Engineering and Computer Science

DECEMBER 2011

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of THANIGAINATHAN MANIVANNAN find it satisfactory and recommend that it be accepted.

_____

Carl Hauser, Ph.D.., Chair

_____

David E. Bakken, Ph.D.

_____

Partha Pratim Pande,Ph.D.

ACKNOWLEDGEMENTS

I thank my advisor Dr. Carl Hauser for his guidance and support for the past three years and also for patiently reviewing each and every chapter of my thesis. I also thank Dr. David. E. Bakken and Dr.Partha Pratim Pande for serving in my committee and reviewing my thesis. I also take this opportunity to acknowledge my parents Mr Manivannan Duraikannu and Mrs Vasugi Manivannan, my sister Karpagam Manivannan, my friends Divya Krishnan, Arun Natarajan, Dilip Raghunathan, and others for being a great support and making this happen. Last but not the least, I am very thankful to National Science Foundation (NSF) for their funding.

EVALUATION AND COMPARISON OF MULTICAST MESSAGE AUTHENTICATION PROTOCOLS FOR USE IN POWER GRID APPLICATIONS

Abstract

By Thanigainathan Manivannan

Washington State University

December 2011

Chair: Prof Carl Hauser

Critical infrastructures such as smart grid require efficient and reliable data delivery services (DDS) for today's and future needs of power system monitoring control applications. Data delivery services have to meet the real time requirements of the smart grid with minimum latency and also provide availability of data at multiple locations using multicast message authentication schemes. All known multicast message authentication approaches carry trade-offs between quality-of-service (QoS) aspects such as added latency, computational cost, and the precise authentication guarantees that they afford.

This paper surveys several multi-cast authentication approaches for use in a power grid DDS based on an experimental evaluation of their latency and computational costs and an assessment of the appropriateness of their authentication guarantees for use in power grid applications. Furthermore, the discussion involves the experimental analysis of different cryptographic primitives for latency and the applicability of different algorithms for data delivery services in smart grid applications. The use of time synchronized protocols such as TESLA are discussed and the latency incurred by them compared to other cryptographic primitives is discussed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

One of the requirements in critical infrastructure such as power grid is to provide timely and secure delivery of reliable data across the power grid. These applications use the PMU data (phasor measurement units) captured from monitoring and control devices. These critical infrastructures prominently use a multicast architecture where the data has to be transferred from a single node to multiple nodes at any specific instant over hundreds of miles with low latencies (20-30 milliseconds)[BBH+11].Authenticating the time-critical multicast data is one of the challenges to provide secure data transfer across the nodes in any critical infrastructure.

A Phasor measurement unit (PMU) is a GPS clock synchronized measurement device which has the ability to sense and measure the voltage and current phasors in the power grid. Each PMU senses the data that need to be sent across multiple control center nodes [HBK09]. These data are yardstick for critical decisions that are carried out in control centers and power stations. Hence, authentication is required to protect the data against impersonation or modifying the data on their way to the control centers which may lead to catastrophic events. As discussed, these are time critical infrastructure and require timely delivery of data across the nodes and again any delay in receiving the data at the destination may lead to useless received data. Using efficient algorithms to minimize the computational cost incurred to encrypt and decrypt the packets at the sender and receiver and achieving minimal latency from the sender to receiver nodes is a major challenge to providing secure PMU data transfer from one node to another node. Minimal packet buffering at the end nodes is necessary to minimize latency, communication overhead, maximize tolerance to

packet loss and resistance to malicious packets. The above mentioned are the major properties that requires attention to provide reliable and secure multicast authentication. However, these properties are inter-dependent on each other and hence improvement in one property may deteriorate the other [QKY+09].

Multicast has gained popularity in the past two decades for use in various applications and the security threats on multicast have grown over years which require the need for secure and efficient multicast protocols.

Multicast is efficient when multiple receivers are subscribed to a group and a single copy of data are forwarded across the router to all the recipients [BCC2000]. Data security is based on confidentiality, integrity, authentication and non-repudiation to the end-user. Multicast communication security is further classified into two major types known as *source authentication* and *group authentication*. In source authentication, the data originates from the source and was not modified on its way to the receiver. There is a chance that the data may be modified or impersonated by any member in the group as if sent by the valid sender.

In group authentication, there is a single key shared by all the group members who are subscribed to a specific publisher which enables only the group members to decrypt and retrieve the message[HM97],[MI97],[Bal95],[WHA00],[WGL98],[MS98].However, there is a possibility for any of the group member to forge the packet and send it to all the group members as if a valid message is sent from the original sender.

Researchers have proposed public key signatures to provide source authentication which involve considerable communication overhead and computation cost. When using digital signatures, each

block of data is appended with a digital signature to prove the authenticity that it is sent from a valid sender. However, the computational cost is high and increases the latency as well for signing and verification at the sender and the receiver [CBB+04],[RSA78].

Further improvements on this scheme have been made to reduce the communication overhead. Only the sender and the receiver can identify the keys and in order to break the scheme, an adversary has to forge all the Message authentication code (MAC) computed with the keys. Instead of appending the whole MAC to each of the message, a specific part of the MAC in terms of few bits or byte is appended. So that they cannot be predicted successfully without knowing the keys. This provides considerable improvement in the overhead thereby reducing the total length of the tag to L*x bits where x is the number of bits of MAC attached to each message [CGI+99].

Another improvement is the ability to reuse same set of keys by different group of senders such that each will hold different subset of keys at any specific moment. Thereby the publisher and subscriber share a sufficient number of keys that are unknown to any bad coalition. In this model, we will have a group of primary keys which derive the secondary keys. The primary keys will be proprietary of the receivers that are trusted and the senders will be provided with the secondary keys. The secondary keys can be derived from the primary keys using any pseudo random function such as keyed hash or block cipher.each receiver uses the corresponding secondary key to verify the MAC and hence no coalition of any number senders can forge other sender's messages.

CHAPTER 2

Survey of Data authentication schemes

2.1 Overview of different multicast authentication protocols

One of the main constraints in multicast environment is to provide individual identity as the PMU data are broadcasted to the subscribers in a multicast cloud. Hence, an efficient mechanism is required to prove the authenticity of a valid sender of the PMU data. Source authentication is one of the burning issues in the field of Internet security. Source authentication assures that the source of the data is what is expected and this can be provided with or without non-repudiation. The term non-repudiation means that the sender of a message cannot later deny having sent the message.

Definitions related to Data origin authentication

Computation overhead: This is the computation cost involved to generate the signature at the sender side and verify the same at the receiving end.

Communication overhead: The overhead incurred due to inclusion of authentication information such as message authentication codes (MAC) or digital signatures when transmitting a packet from sender to the receiver.

Sender delay: The time delay between the processing of data packet by adding authentication information and sending out from the sender node.

Receiver delay: It refers to the delay from the time the packet is received at the end node to the time it takes to authenticate and retrieve the original message.

Robustness against packet loss: The packets are assumed to be transmitted over a lossy channel and all the packets should be verified at the receiver even in case of packet loss

Authentication probability: The probability of packets sent from the publisher successfully verified at the subscriber in the presence of packet loss is known as the authentication probability.

Receiver side buffering: The maximum number of packets required at the receiver end to compute the robust authentication information.

Authentication information size: This corresponds to the amount of overhead appended to the packets to authenticate the message.

Time Synchronization: This corresponds to the need for the sender and receiver to be synchronized with each other to receive the packets and authenticate them.

2.2 Data origin authentication without non-repudiation

For a given message M, the sender and the receiver share a secret key and authenticity of the message is proved using the secret key. In case of multicast authentication, the same secret key is shared by the group of receivers and the sender of the message. When the sender sends the message, the message authentication code is computed using the key and is appended to the

message. The corresponding messages along with the MAC are forwarded to the subscribed group members and each receiver computes the MAC code and verifies with the sender. But, the major problem is that any of the receivers can forge as if it is sent by a valid sender. Any receivers can multicast data to the group as the secret key is shared by all the members of the multicast cloud. In order to solve this issue, asymmetric authentication is preferred where the sender utilizes a public key and the receivers make use of common public key to verify the authentication of the message [CBB+04].

In secret-information asymmetry, a share of secret is known to each receiver which is sufficient to authenticate the sender message. The whole secret message is known only to the sender and it's not possible for the receivers to forge the message. With minimal authentication information shared by senders, the receivers can only authenticate the message. It is infeasible for it to forge a message as if send by a valid sender. However, the minimal authentication information is available to each receiver and is possible only to acknowledge the receiving message. In case of time-asymmetry, the sender and the receiver are time synchronized with a key delay in order to send the key to authenticate the message. The time delay in which the key is sent between two ends varies periodically reducing the possibility of an attacker to forge a message on behalf of the sender.

Source authentication without non repudiation can be provided by the protocols such as RABIN M [Rab98], erasure codes [PCS03], TESLA [PCS+00], BiBa one-time signature protocol [Perrig01], multi-level μ-TESLA [LN04], localized broadcast authentication [SQ06], broadcast authentication protocol with time synchronization and quadratic residue chains

[Groza07]. These schemes can prove the authenticity of the received message but doesn't guarantee that the message originated from the trusted source.

In order to provide non repudiation and to prove to a third party that the data received and sent by the source are the same, many schemes are established such as signature amortization techniques [PCS02], online/offline signature scheme [GR97],[GR01], one time signature [Lam81,Rabin78], tree chaining [WL98], augmented chain [GM01], A2CAST [Challal et al], RLH protocol [HBC06], HMSA [JGX+07], EMSS [PCS02], SAIDA [PCS02] and graph based authentication of digital streams [ZSW05].The taxonomy of different data origin authentication schemes were discussed in figure 2.1.

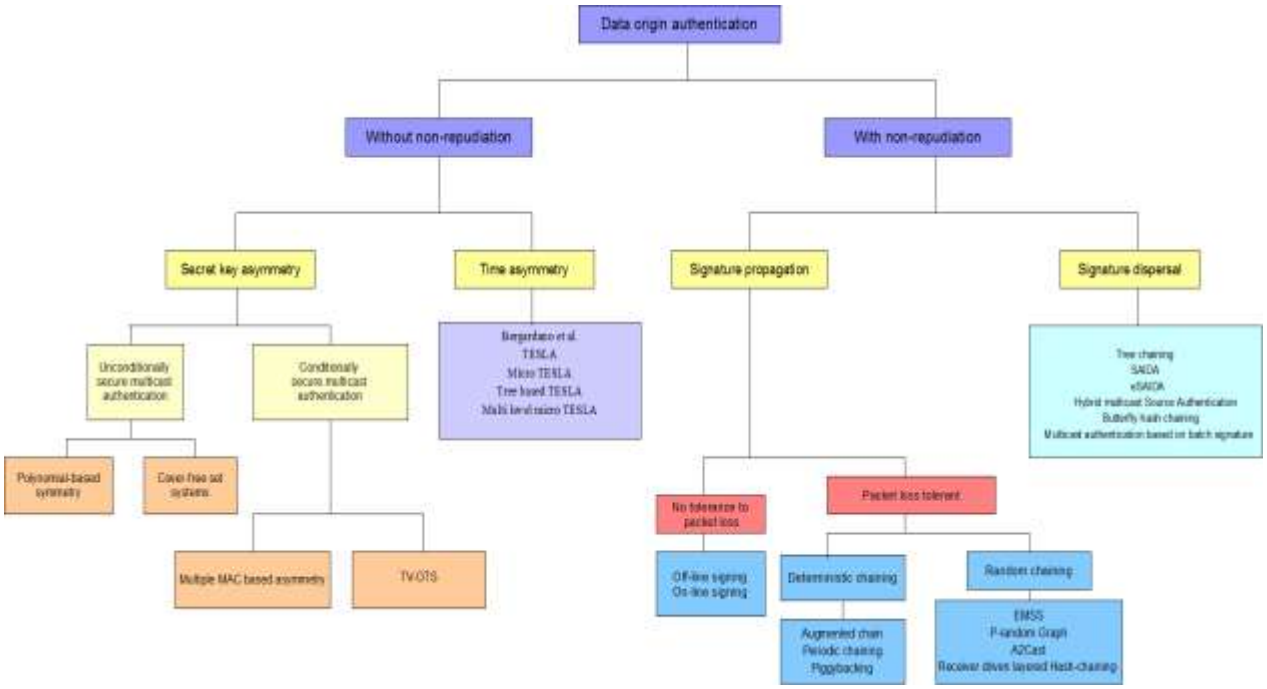

Figure 2.1: Taxanomy of different data origin authentication schemes

The more straight forward way to provide source authentication is by using hash and digital signatures. However, these are computationally expensive to sign each packet and also results in increased communication overhead.

2.2.1 Secret Information Symmetry

Secret information symmetry can be classified as unconditionally secure multicast authentication and conditionally secure multicast authentication. The security of the multicast authentication scheme are not based on unproven assumptions, on the other hand, they are based on the fact that the absolute security can be obtained by making sure that the attackers are not provided with enough information. This scheme is based on information theoretic strength rather creating a complex issues based on computational complexity. In order to secure encryption mechanism, the keys should not be reused which leads to the need for large keys when using this scheme. Furthermore, use of signatures is not possible while using unconditional system since they rely on computational assumptions based on unproved assumptions related to hardness of problems.

Desmedt et al. came up with an unconditional multicast authentication scheme based on polynomials where the signature scheme is based on *t* out of *l* multicast authentication with *t* shareholders combined together to authenticate a message. However, *t-1* out of *l* senders cannot commit a substitution of impersonation attack as they will not have enough information to authenticate the message. Now the source sends a message m(x) of degree t to each of the receiver a share of the polynomial such that each of the *t-1* receivers gets a unique share. The model is considered to be secure since the substitution or impersonation is not possible in the

scheme and furthermore *t* shares are required to reconstruct the original message.The protocol incorporated in this scheme is described as follows.

- A large prime number p is chosen by the sender such that the selected number is larger than the number of possible messages. All the following operations are done in the finite field Z_p(integers modulo p).
- P0(x) and P′(x) of degree k are created for each message M.
- The polynomial of the sender message is generated such that message M: M (i) = P0 (x) + M * P ′(x) and send to each receiver.
- The sender transmits P0 (i) and P′ (i) privately to each receiver (i).
- At the receiving end, the authenticity of each message is verified using already received function AM (i) = P0 (i) + M * P ′(i).

This protocol by Desmedt et al is an efficient scheme providing multicast data origin authentication with unconditional security. It is unsusceptible to packet loss since each packet carries its own authentication information and is verified individually as soon as the packet is received. The major drawback behind the scheme is that the time taken to generate and share the polynomials across the receivers increases the latency to a great extent which makes it impractical to use this scheme in real time applications.

Safavi et al. proposed few ideas to extend the Desmedt et al work on multi-receiver A-codes. Multi-receiver A-codes are constructed using traditional A-codes in which the sender shares a common key with each receiver and constructs n codewords for each receiver. The sender concatenates all codeword's with the message and multicasts the result. Now each receiver verifies its own codeword and authenticates the message. Desmedt, Frankel and Yung (DFY)

polynomial scheme for multi-receiver A-codes were generalized to authenticate multiple messages instead of a single message. Desmedt et al work utilized 2t polynomials to authenticate t consecutive messages on the other hand, the latter scheme requires only t+1 polynomials for the same number of receivers. Hence, the key storage required for the sender is reduced from ((w+1) k log q) to ($\frac{w+1}{2w}$ k log q) and the receivers ((w+1) log q) are reduced to ($\frac{w+1}{2w}$ log q) respectively where q determines the size of the key storage and length of the authentication tag. However, the length of the authentication tags for both the construction remains the same [SW98], [CBB+04].

This extended work also provides a scheme introducing a new construction for multi-receiver A-code when the number of receivers or the size of the source is large. The scheme is implemented by combining arbitrary A-code and a special combinational structure called cover free family. The main idea of cover-free family is that given a set of keys used by the sender to authenticate messages, the group of receivers with their subset of keys should not collaborate and cover the subset keys of a group member.

In this way, the (k,n:w) multi-receiver multi-message authentication code is designed to transmit w-1 authenticated messages to n receivers in such a way that the opponent nor the k-1 receivers cannot impersonate or do substitution attack to any of the other receivers. Secondly, each receiver can verify the authenticity of the message independently. This paper, also presents a discussion regarding an improved version of multicast authentication with dynamic sender in which the transmitter identity is kept secret and is not determined beforehand among the other nodes in the network [SW98].

Obana and Kurosawa [OK01] proposed the optimum scheme with tight bounds to implement MRA codes which uses a special pair of orthogonal arrays named TWOOA to

construct a (k,n:2) multi receiver message authentication code. In an MRA code, the sender wants to authenticate a message for group of receivers such that each receiver can verify authenticity of the received message. It is assumed that it is infeasible for a fraudulent receiver to construct a message on behalf of the sender. R.Fujii et al further extended the use of TWOOA and generalized TWOOA to construct a (k,n:w) multi-receiver multi-message authentication code which exceeds the implementation done by Safavi et al in number of receivers and authenticated messages. The discussion about this is beyond the scope of the thesis and the readers are encouraged to reference [FKK96] for more information.

2.2.2 Conditionally secure multicast authentication

Conditionally secure multicast authentication schemes rely on the fact that the attacker does not possess sufficient resources to solve complex computational algorithms in order to guess the secret keys and hence forge authenticate messages within real time deadlines. The assumption behind is that the valid MAC for a message cannot be predicted by the hacker and is known only to the original sender and receiver.

Canetti et al. proposed a scheme to append multiple MACs in each message to achieve conditionally secure multicast authentication. In this scheme, the sender picks a message M and appends l different MACS computed using l different keys $K_1 K_2 \ldots . K_l$ such that each of the t receiver is securely provided with a subset of $\{K_1 K_2 \ldots . K_i\}$ keys from $\{ K_1 K_2 \ldots . K_l\}$. Each data packet can be represented as

Message Packet: Message M || MAC (K_1, M) || …..|| MAC (K_l, M)

At the receiving end, each receiver verifies consistency of received MAC with the subset of K_i MAC. The authenticity of the received packet is confirmed if the match is found, otherwise it is rejected. This scheme is named as conditional multicast authentication because for the intruder to forge the message and prove that it is sent from a valid sender, it needs to have *l* keys from a coalition of w receivers. The assumption is that right choice of receivers subset of keys makes it least possible for the w colluding bad members to determine the original keys possessed by the good member.

The main drawback of the approach is that the number of keys as well as the packet overhead increases with number of colluding members w. This limits the use of this scheme in large networks such as Internet where the number of adversaries may be abundant.

Canetti et al came up with an efficient scheme by lowering the communication overhead by using four times as many keys as in the basic structure such that one can make sure that w colluding receivers will not know log (1/q) sender keys. This way each key can produce a single bit output instead of whole MAC thereby decreasing the communication overhead. Each sender is given an upper-bound probability $l = 4e^{w \ln(\frac{1}{q})}$ keys where w is the size of largest colluding receivers such that the probability of w colluding receivers can recover the subset K_i of another group is upper bounded by probability q and each receiver is given a key K_i with probability $\frac{1}{w+1}$ .Thus proposed scheme reduces the size of MACs to a single bit as output thereby reducing the authentication information to *l* bits. The major advantage of this scheme is it is tolerant to packet loss.

2.3 Data Origin authentication with non-repudiation

Data origin authentication and non-repudiation can be achieved by the use of digital signatures where the sender signs the packet using its private key and multicasts across all the receivers. The receivers use the public key provided by the valid sender to verify their authentication. However, as discussed previously, the computational complexity of the digital signature used to sign and verify leads to impractical solution in an application that should adhere with real time deadlines.

2.3.1 Signature propagation

New methods have been proposed to provide authenticity and non-repudiation without the use of digital signatures. Instead of signing each packet individually, a small piece of information in the first packet is signed by the valid sender which carries authentication information required to prove the validity of the second packet and hence, piggybacks the authentication information of following packet. In this way, a chain structure of interdependent packets where the first packet carrying the signature to prove the authenticity of all the future packets is implemented and is called as *signature propagation*.

2.3.1.1 Offline-chaining

Gennaro and Rohatgi proposed a model where the sender identifies the entire stream in advance off line. In this protocol, the entire stream is divided into blocks and the sender computes the hash of a block. The computed hash is appended to the end of the subsequent block to form a chain. Finally, the sender signs the hash of the last block. At the receiving end, the signature of the first block is tested and the origin of streams' data is verified. With the verification of the signature for the first block, the receiver identifies the expected first block hash. At the end, the

entire stream of data is received and the hash of the incoming block is compared with the hash of the previous block to authenticate all the blocks in the stream [GR01].The offline chaining scheme is illustrated in figure 2.3.1.1.

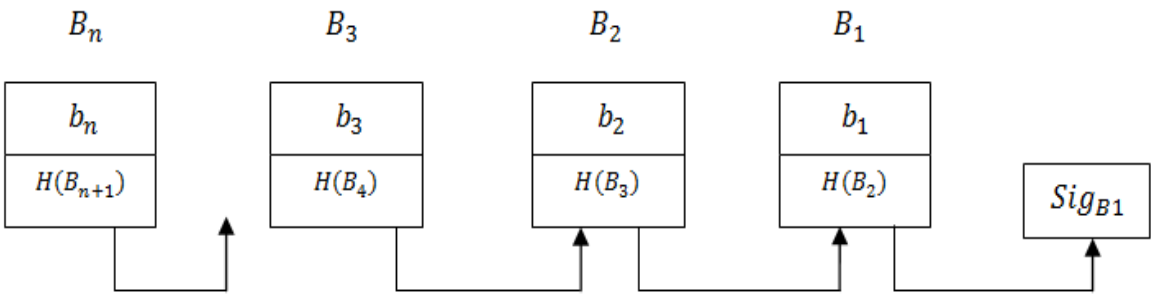

Figure 2.3.1.1: Offline chaining scheme

The major advantage of this scheme is that entire stream is authenticated with a single signature and n hash computations required at both sender and receiver end. In addition, no buffering is required at the receiver end. However, the major drawback is that the solution is not susceptible to packet loss. The loss of one packet leads to break of authentication chain and thereby all the subsequent packets cannot be authenticated [GR97].

2.3.1.2 On-line chaining

Gennaro and Rohatagi introduced on-line chaining in order to overcome the issue with off-line chaining where the sender must identify the entire stream in advance [GR97, GR01]. In this scheme each data packet is split into n blocks and each block carries one-time public key signature along with the hash based on the one time signature of the previous block. At the receiving end, the receiver authenticates B_K from one time signature received with the block and

the one time public key received from the previous block B_{K-1}.The online chaining scheme is illustrated in figure 2.3.1.2.

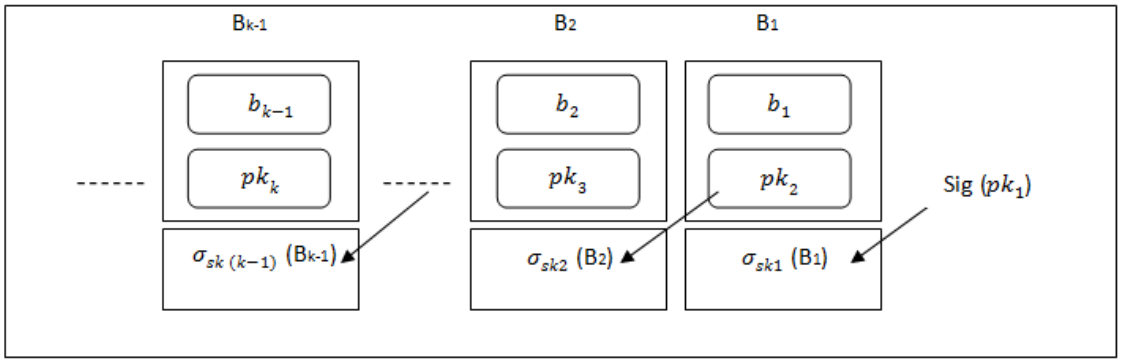

Figure 2.3.1.2: Online chaining scheme

In this way, the sender can sign each block on the fly without waiting for them to identify the entire stream of message that is sent to the receiver. The use of one time public key in each block increases the computation overhead at block level which is not immune to packet loss. The loss of block in the middle will interrupt the verification for rest of the blocks.

2.3.1.3 Efficient Multi-chained Stream Signature (EMSS)

Perrig et al proposed a probabilistic approach with the combination of hash functions and digital signatures to authenticate packets named as EMSS (Efficient Multi-chained Stream Signature). EMSS overcomes the drawback pronounced in stream signing technique which is not robust against packet loss. In this scheme, the hash of each packet is stored in multiple locations or in random to provide redundancy in case of packet loss so that the authenticity of chain of packets is verified using the redundant hash of the message from another packet at different location. There is a chance of loss of packet which contains the signature and results in failure to authenticate all the packets before the break point. This can be avoided by sending the signature

packet for multiple times after a certain delay so that packet loss can be correlated. The block diagram of EMSS scheme is shown in figure 2.3.1.3.

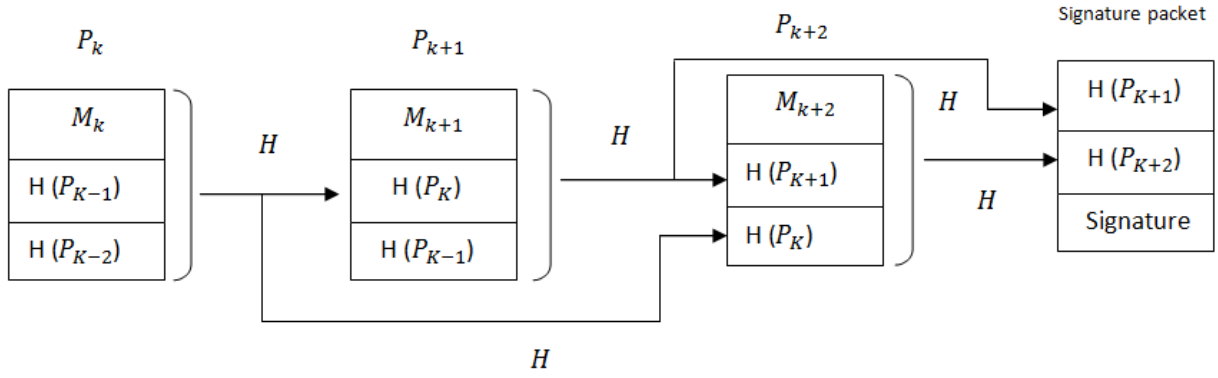

Figure 2.3.1.3: EMSS scheme

The three drawbacks in EMSS are the use of number of signature packets, number of hashes contained in the signature packet and the number of hashes in the data packet which is always greater than one to provide the required redundancy to ensure packet loss tolerance. The increased space overhead is unavoidable but can be made more efficient by using erasure codes [PCS03]. The above mentioned drawbacks can be overcome by using a new scheme named SAIDA (Signature Amortization using Information Dispersal Algorithm).

2.3.1.4 Signature Amortization over information dispersal algorithm (SAIDA)

In SAIDA, we use IDA to amortize a single signature over blocks of packets thereby reducing the size of authentication overhead and provide better space efficiency. SAIDA is an efficient authentication scheme to provide high robustness against packet loss with optimum communication overhead. It achieves high verification probability when compared to other schemes such as EMSS, augmented chain technique and tree chaining making it useful in

multicast applications. The only drawback with SAIDA is that the packets are generated in real-time environment and immediate broadcast of the packets are crucial with critical time constraints [PCS02].

Park et al proposed a stream authentication scheme in which each stream is divided into blocks and hashes of each packet is computed in a block. The information dispersal algorithm (IDA) [Rab89] is then used to compute the authentication information with added redundancy. The mechanism disperses n hashes of n block packets and the block signature is split into n pieces such that the n pieces can be reconstructed at the receiving end even if n-x pieces are lost.SAIDA architecture is illustrated in figure 2.3.1.4

The authentication procedure for SAIDA procedure is as shown as follows,

- Let us divide the stream of packets into i blocks of n packets such as B_1, B_2, B_3 ,…, B_i
- An hash function H is used compute hash of n packets each block such that f_i = H (P_1^i) || H(P_2^i) || H(P_3^i) || ….|| H(P_n^i) where P_n^i is the nth packet in block B_i.The sender signs the block B_i by signing F_i (the hashes concatenation).This block signature is denoted as S_i.
- For both the hashes f_i and signature S_i, IDA algorithm is applied to get IDA pieces f_i^j and S_i^j where j varies for n values.
- Each packet P_j in the block B_i is appended with corresponding IDA-pieces, f_i^j is the jth IDA-piece of f_i and S_i^j is the jth IDA-piece of S_i.
- At the receiver side, the IDA algorithm is used to reconstruct the authentication information and verify the packets. This scheme can tolerate packet loss until n-x authenticated packet loss.

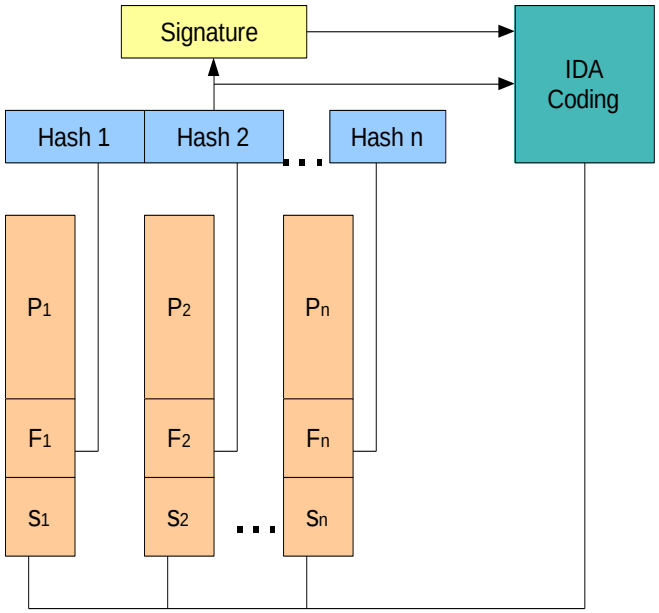

Figure 2.3.1.4**:** SAIDA Architecture

SAIDA provides a balance between the authentication overhead and packet loss. For instance, when more redundancy is added to the packets, packet loss attains more tolerance thereby increasing the authentication overhead. On the other hand, the authentication overhead is decreased to a large extent which results in decrease in tolerance to packet loss. The major drawback with the scheme is the use of information dispersal algorithm which provides tolerant to packet loss. But, at the same time, delay latency at the sender and receiver is increased due to IDA processing. In spite of these drawbacks, the performance in terms of verification probability is considered to be best. Due to significant delay in IDA processing at both sender and receiver, this cannot be applied when data has to be distributed in real time applications.

eSAIDA is an improvement over SAIDA proposed by Yongsu et al [PC03, PCS04]. In case of SAIDA, hash is computed for each packet in a block while in eSAIDA, each adjacent packet carries a single hash to reduce the overhead in the SAIDA scheme. However, the major

drawback of this scheme is that the packet will remain unverified if the couple is not tested correctly leading to lesser verification rate compared to SAIDA and indeed provides lesser communication overhead [PC04, HM10].

2.3.1.5 Periodic chaining

Periodic chaining scheme works similar to EMSS where redundancy hash-chaining is used so that the hash of each packet is linked to several target packets randomly. However, in the case of periodic hash chaining, each packet hash are linked to the multiple packets in a deterministic way. This deterministic design approach optimizes resistance to burst loss. The main motive of the periodic hash chaining is to maximize the size of longest single burst of loss that the authentication scheme can withstand using the periodic hash chain. To construct a topology of hash-links between packets, the sender needs to buffer some packets to create hash-links between them, denoting the size of this sender's side packet buffer by p packets.

The proposed topology of periodic chaining depends on the possible sender's buffer size (p).

- If there is no buffering at the sender size (p = 1), then the proposed topology is denoted as C_a. The target packets of a packet P_i are P_{i+1} and P_{i+a} and the last packet P_n is signed at the sender side.
- If the buffer size is p > 1 at the sender, then the proposed C_a augmented chain topology is denoted by $C_{a,p}$ .The $C_{a,p}$ topology relies on C_a topology by inserting p-2 packets recursively between two packets of the C_a topology.

The burst packet loss due to bulk transmission of data can be controlled by using this scheme. However, the main drawback is that the sender needs to buffer p packets before sending them

which increases the end-to-end latency. This makes it impracticable to be used in live-broadcasting and real-time applications.

2.3.1.6 Piggybacking

Miner and Staddon [SJ01] came up with a solution named piggybacking to resist multiple loss bursts designed by classifying the priority of each class. In this model, hash chaining is implemented in such a way that the increase in priority of the class will increase the redundancy to provide hash-chaining of packet to that particular class thereby providing greater resistance against burst losses. The packets from the source to destination are split into priority classes ($C_0, C_1, C_2 \ldots C_r$). The first class will have high level of tolerance to the packet loss such that they are regularly placed throughout the stream to minimize the probability to be lost in a burst loss of packets.

Piggybacking scheme is particularly attractive when the sender wants significant control over the tolerance of each packet. The major drawback with this scheme is that the redundancy degree within each class is lower-bounded by the maximum number of bursts that could be tolerated, but in practice this information is not easy to acquire. Moreover, the proposal requires buffering at both the sender and the receiver which increases the latency as the signing and the verification process are delayed. These results which are piggybacked are not used in real time applications such as live video or multicast publish-subscribe applications.

2.3.1.7 Augmented chain technique

In another approach aiming at providing optimal resistance to packet loss, Golle and Modadugu [GM01] proposed augmented chain technique which is designed in a systematic way to insert

hashes in strategic locations to provide better tolerance for packet loss. In this scheme, each packet contains the hash of the subsequent packet as well as the previous packets. The augmented chain is constructed in two phases. In the first phase, hash of the message packet is appended to previous and forthcoming packet. In the second phase, p-1 packets are inserted between each packet in first phase and the hash value is stored in adjacent packets inserted in the first phase. The original stream of packets is divided into blocks and each block applies the augmented chain technique which would reduce the verification delay to great extent. This scheme has the resistance to random packet loss and optimal resistance to burst packet loss [GM01]

During the transfer of streams of packets from source to destination, there is a probability for loss of packets in transit and hence requires retransmission to recover the original data. However, the retransmission in real-time systems will increase the end-to-end latency. In order to overcome this limitation, the forward error correction technique is used which helps in increasing the robustness to the packet loss but also increases the space overhead to a large extent. For authentication schemes, the two popular erasure codes are Tornado codes [LMS+97] and Rabin's information dispersal algorithm (IDA) codes [Rab98]. The former can encode/decode very rapidly with the large number of segments to encode in linear time. In IDA case, some amount of redundancy is added to the source file F and the result is split into n pieces and transmitted. Reconstruction of file is possible with any combination of m pieces where $m < n$. This scheme is space optimal and the space overhead can be controlled by the parameters n and m that would decide the required amount of redundant data added to the source file.

2.3.1.8 A^2cast

A^2cast, a new adaptive and efficient source authentication is introduced which can tolerate packet loss like EMSS as well as guarantee non repudiation for multicast streams similar to SAIDA. This protocol integrates the functionality of SAIDA and EMSS. This protocol uses adaptive hash-chaining technique to amortize single digital signature over many packets which allows us to save bandwidth and improves the probability that a packet be verifiable even if some packets are lost. In this scheme, the hash value of each packet is included in multiple subsequent packets chosen randomly to provide redundancy and multiple packets are signed using a single signature. This protocol provides tolerance to burst packet loss patterns as well as reducing the communication overhead by signing a group of packets with a single signature [Challal et al].

2.3.2 Signature Dispersal Introduction

The straightforward way to provide non-repudiation in a multicast topology is to append hash and signature in each packet to authenticate the message at the subscriber end. This is an inefficient solution as the communication overhead is very high. In block signature schemes, all the hashes in a particular set are signed by a single signature which results in reduced communication overhead, but on the other hand susceptibility to packet loss.

Signature dispersal schemes are introduced as a practical solution in which at most k out of n packets are assumed to be lost, where k<n. The proposed solution reduces the amount of authentication information carried by each packet and in addition, the reasonable assumed threshold value reduces the packet loss tolerant rate. This scheme is based on processing

authentication information and dispersing it throughout a set of packets. The processing is carried out in such a way that even though some amount of processed information which is transmitted is lost (assuming it's less than the threshold value), all the authentication information can be recovered from the received data.

2.3.2.1 Tree hash chaining

In tree hash chaining [WL98], the hash value is computed as the root node of an authentication tree. For example, consider a block of eight packets $P_{1,}$, $P_{2,}$,.. P_8 with hash value $H_{1,}H_{2,}$,... H_8. The packet digest forms the leaf node of the binary authentication tree, the parent of hash are formulated as ($H_{1,}$, $H_{2,}$) = $H_{12,}$ and ($H_{3,}$, $H_{4,}$) = $H_{34,}$ and ($H_{12,}$, $H_{34,}$) = $H_{14,}$ and so on. The sender now sends the first packet containing signed hash of the root along with packet ID, hashes of siblings and group signature containing signed hash of the root.

In this scheme, each packet is individually verifiable and can tolerate the packet loss although *i-1* packets are lost out of *i* packets. However, the major drawback of this scheme are utilizing large communication overhead since each packet has to carry signature of the root, hashes of sibling of packet from the current path to the root and requires buffering at receivers to authenticate the packet. In addition, the use of digital signature with each block at both sender and receiver further increases the computational overhead. The tree chaining scheme is illustrated in figure 2.3.2.1

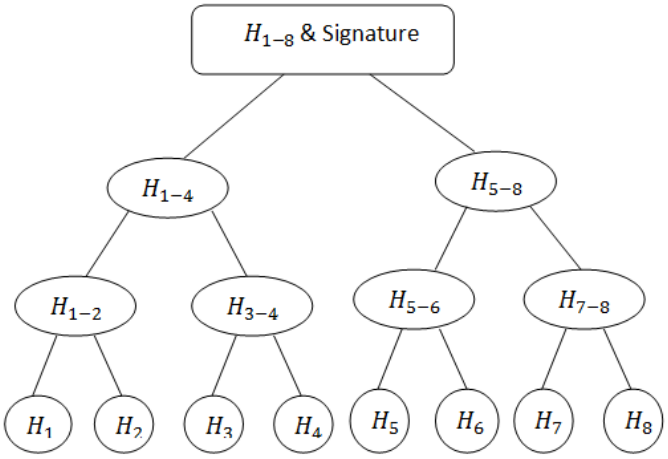

Figure 2.3.2.1**:** Tree chaining scheme

The major issue with the use of hash tree is the high communication overhead incurred by the use of m hashes and m signatures where m-1 signatures are unused since the goal is to use the root signature of the first block for verification. On the other hand, the use of hash chain is intolerant to packet loss and the data loss will interrupt receiver to authenticate future packets from the sender. Furthermore, the latency incurred at the receiver is increased since the inoperative packets need to be buffered and the process is extended until the entire packets expected before they are received and verified.

2.3.2.2 Hybrid Multicast Source Authentication (HMSA)

HMSA is formulated by efficient combination of hashing tree and hashing chain with systematic use of hashing chain protocol to reduce the communication overhead incurred due to packets carrying useless signatures in each packet. In HMSA, the sender calculates the root hash of each block using hashing tree and then sends the packet carrying log n hashes of siblings from all the nodes in the current packet's path to the root and the root hash of the next block. At the receiving

end, the root signature of the first block is verified followed by the root hash of the second block being verified and similarly the origin of all packets verified in the same way [JGX+07].Each packet is authenticated immediately on arrival at the receiver and no buffering is required thereby improving latency.

In this way, the HMSA needs only one signature and requires $log_2 m + 1$ hashes thereby reducing the communication overhead and computation compared to the hashing tree. On the other hand, the above attributes are much higher than the hashing chain protocol.

2.3.2.3 Receiver driven Layered Hash-chaining (RLH)

In yet another scheme named Receiver driven Layered Hash-chaining for multicast data origin authentication protocol (RLH), redundant hash-chaining scheme is used to tolerate the packet loss. To reduce the communication overhead created due to signature scheme, hash chaining is applied based on amortization of single signature over multiple packets and is implemented by hash-linking the current packet to another stream of previous packet. However, the major disadvantage is that the scheme is not packet loss tolerant.

RLH proposes the use of redundant hash-chaining scheme to tolerate the packet loss by providing different layers of redundant data and implementing dynamic behavior in selection of the redundant hash-chain according to the degree of packet loss between sender and receiver. It consists of a basic layer that carries the payload data packets with a minimal hash-chaining redundancy degree. This layer is used in combination with different amounts of redundant hash-chains [HBC06].

Initially, the basic layer joins the multicast group and assures robustness to a certain degree of packet loss ratio whereas the receiver periodically calculates the actual packet ratio and decides whether to join an extra layer to improve the packet loss tolerance. This decision is made by a function known as update membership which is defined as a packet loss ratio parameter. The advantage of redundant hash-chaining scheme is that it is localized to the subnet. For example, if we take a large network with number of subnets, the degree of packet loss at each subnet varies in accordance to the traffic and various parameters that affect that specific subnet. Hence, redundant hash-chains required for a specific subnet cannot be applied to all the subnets for the entire network and may increase the unused redundant data to a large extent. The scheme efficiently utilizes the subscription policy for all subnet in such a way that each subnet joins the group in accordance with the packet loss ratio at that specific subnet.

2.3.2.4 Butterfly hash chaining

Zhang et al [ZSW05] proposed a scheme based on butterfly-graph supporting stream authentication for lossy networks under the assumption that the stream of packets can be lost in both random and burst ways. The entire stream is divided into n number of blocks with each block containing M packets such that M = N (Log_2N+1).

The definition of butterfly acyclic graph states that, it is a directed acyclic graph (DAG) with a single signature packet S and the total number of packets is given by M packets. The total number of packets is divided into (Log_2N+1) stages with each stage having N packets. The packet is denoted as P(s,j), where s ∈ {0,1,…, Log_2N } indicates the stage and j ∈ {0,1,…, N-1} indicates the packet in a stage. In this graph, there exists a directed edge $\underset{e}{\rightarrow}$ *(P (s_1,j_1), P (s_2,j_2))*

from packet P (s_1,j_1) to packet P (s_2,j_2) the following condition $S_1 = S_2 + 1$ and $j_1 = j_2$ is met. There also exists a directed edge from all packets in stage0 to the signature S.

In the butterfly authentication graph, each directed edge is realized by appending the hash of the packet P(s_1,j_1) to the packet P(s_2,j_2). Consider an example as shown in figure 2.3.2.6 in which N = 3, with 4 stages and 8 packets in each stage. The hashes of all packets at stage 0 and the signature packet are contained in the signature packet S. All packets from stage 0 to log N-1 have two hashes. The last stage does not contain any hash. The block diagram of butterfly hash chaining scheme is shown in figure 2.3.2.4.

Consider, each hash as h bytes, each signature has g bytes and each block has M = N(Log_2N+1) packets, hence, the communication overhead per packet incurred due to use of this scheme is given by 2h - $\frac{h}{\log N+1} + \frac{g}{M}$. The communication overhead is considered to be quite low with one signing operation for the whole block and one hashing operation per packet.

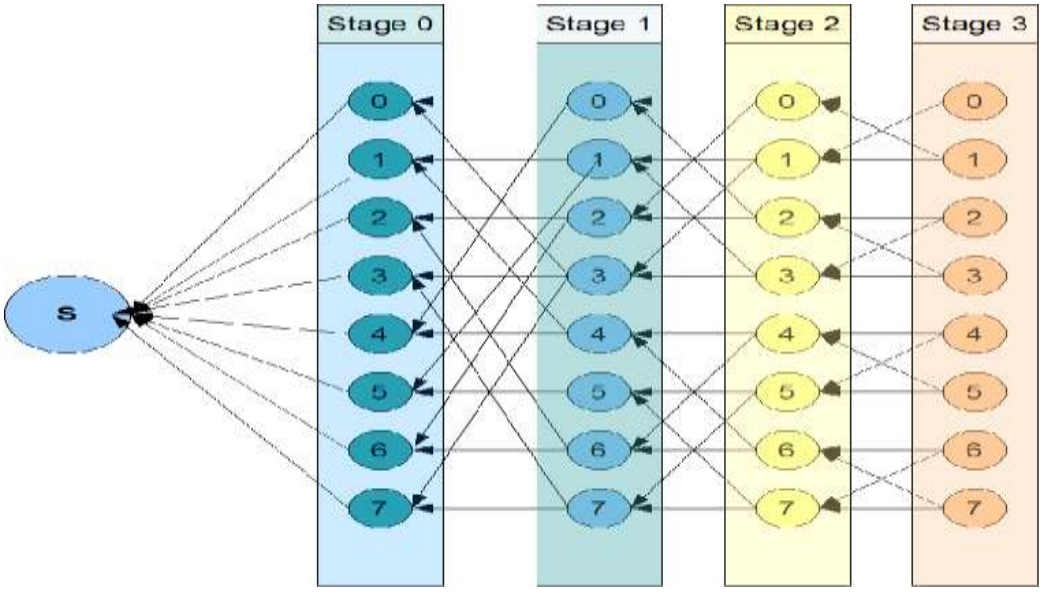

Figure 2.3.2.4 Butterfly hash chaining

The drawback of this scheme is high sender delay M incurred due to the hash computation and the completion of block signing operation before the first packet is sent. However, the receiver delay is quite low and need only a small buffer at the receiver.

[ZSW+07] proposes an improved framework based on generalized butterfly graph. The main aim of this scheme is to improve the verification probability of packets according to the design based on total number of packets, packet loss rate, and overhead budget. The scheme also comes up with a new evaluation metric to measure the effect of authentication method for the packet loss rate known as Loss-Amplification-Factor (LAF).

2.3.2.5 MABS: Multicast Authentication Based on Batch Signature

Several authentication schemes has been discussed so far based on block based authentication and (m, n) coding scheme. A new scheme based on batch signatures is proposed by [ZF06, ZF07, ZZF10] in which multiple packets are authenticated simultaneously as a group. The main motive behind this scheme is to improve the computational complexity of verifying operation by validating the signatures in bulk rather than individually.

In this scheme, each packet is signed by the sender individually. At the receiving end, the packets are collected as a batch of n packets $P_i = (M_i, S_i)$, where M is the message, S is the corresponding signature and i varies from 1, 2, 3…n. The n packets are provided as an input to the verification algorithm named Batch signature which receives packets (P_1, P_2, P_3 .... P_i), hence, the output is obtained as true if the all the signatures are verified. On the other hand, the output is false even if a single signature from that batch cannot be verified. In this way, the

computation complexity involved in packet verification as well as the authentication latency due to packet buffering in block based scheme is eliminated.

In this scheme, there is a possibility to forge the packets by the attacker in order to enable DoS attacks. However, this can be eliminated by attaching unique marks to the packet using Merkle tree [Merkle80] and classifying the received broadcast streams into disjoint sets based on marks. Thus, we can ensure that the forge packets does not mix-up with the packets from the valid sender. The main drawback of this scheme is the use of n signatures and O (n log n) hashes to provide better security. To tolerate packet injection, Zhou et al proposed an MABS-E which is a combination of basic MABS-B and packet filtering mechanism [ZZF10].

2.4 Time Asymmetry Introduction

Authentication and non-repudiation of multicast packets with minimum latency has always been a challenging problem in real time applications. Time asymmetry scheme is based on setting up threshold values for lifetime of keys that are used to authenticate the multicast packets upon reaching the receiver. Assuming a condition that, the attackers eavesdrops the key and use it to compute the MAC by impersonating the message as if sent by the valid sender. Then, the receiver will simply reject the packet as the time taken to receive the specific key is expired. There is always a considerable delay incurred if the impersonator changes the packet and sends it as if a valid sender. Time symmetry scheme calculates the time delay and considerable network delay that are incurred in real time applications. The scheme also computes the maximum time bound within which the keys should reach the receiver in order to prove the authentication that it is sent from the valid sender. Also for each packet unique keys should be used. The secure transfer, key module change and maintenance becomes complex for power grid applications. To

overcome this, one way chain is considered as a solution which deals with authentication and generation of keys and tolerance to packet loss. However, loss of few keys in transit may lead to authentication of corresponding packets using the later received keys. [JP05].

An effective broadcast/multicast authentication protocol design for power grid data dissemination must cater to the below requirements:

- The overhead involved in generation and verification should be small to reduce the latency.
- There should not be buffering of messages at the sender for signing operation nor at the receiver for verification.
- Each message should be able to be authenticated instantly without waiting for future messages which will piggyback authentication information.
- All the broadcast/multicast should provide high resistance to packet loss and should have mechanism to retrieve lost packets without retransmission.
- The scheme should be scalable to large number of receivers applicable to any topology of publishers/subscribers.

The Guy Fawkes protocol proposed by Anderson et al [ABC+98] introduced a chaining mechanism that appends keys with messages to provide non-repudiation. The concept of one time password proposed by Lamport [Lam81] used in various applications is extended in Guy Fawkes to allow the keys to be refreshed and set timeliness for expiry of a shared key. However, this scheme exhibited delay related issues and constraints when used in Broadcast environment. Bergadano et al [BCC00, BCC2000] proposed a solution using hash chains as well as provided solutions to delay related issues previously faced in Guy Fawkes protocol.

In this scheme, each packet uses a collision-resistant hash function to generate a MAC using a one-way key chain. The use of one way key chain helps to recover lost keys on the way of transit to the receiver. This scheme requires the sender and all the multicast receivers to be synchronized with each other. In order to attain the synchronization, the sender session id, sender processing delay and sender time is signed with hash of the secret which is then sent to all the receivers during the initial setup phase. The MAC along with the message is multicast to all the receivers and the key required for verifying the MAC is sent to the receiver after a delay. The challenging aspect in this scheme is the delay calculation for each multicast receiver since each recipient can have different delay and there is a possibility for fraudulent receiver to receive the key early and forge the message as if sent by a valid sender. In order to avoid the forgery, the sender is synchronized with all the receivers and key disclosure delay is selected to be as large as possible so that all the receivers have received the packets and are authenticated with their respective keys. On the other hand, when the packets arrive very late more than the delay time, the receiver assumes that the packet has been forged and rejects the data packet

Let x be the random seed using the hash function h to create hash of a seed $x_k = h^k$ (x) and the maximum sender delay processing time is given by $D_s \geq 0$ at the sender. If we consider the sender side session id as S_n, and starting time as S_t, the signature of a message A is given by $Sig_A (S_n , S_t, x_k, D_s)$. This is the only signature which will provide the authenticity for the chain of messages A_i sent by the sender.

At this stage, the first message sent by the sender will multicast with the corresponding MAC h^{k-1}(x) of A_1 and the message A_1 at sender time S_t. The second message will be sent with

a time delay T with respect to S_t (i.e., S_t + T) to provide sufficient time for the first message to reach all the receivers such that MAC h^{k-2}(x) of A_2 at time S_t + T and so on.

The keys from the sender are multicast to all receivers depending on the longest delay latency incurred including all factors that can be experienced from the sender to the receiver given by the expression, delay = (latency+D_s + D_r + R) where $D_r \geq 0$ is the receiver clock precision, R is reliability parameter , latency is the maximum propagation time . The expression to multicast the first key $x_{k-1} = h^{k-1}$(x) is given by S_t + T + delay. The expression to multicast second key is given by $x_{k-2} = h^{k-2}$(x) is given by S_t + 2T + delay.

At each receiver, first the signature Sig_A (S_n , S_t , x_k, D_s) is received and its authenticity is verified. If the signature is not authentic, no further action will be taken to check the authenticity of future packets. On the other hand, if it is authentic, then each MAC computation is verified using MAC x_{k-i} of A_i , A_i and x_{k-i} in orderly manner such that hash of h(x_{k-i}) should be appended in x_{k-i+1}. The verification of MAC alone is not sufficient to authenticate a particular message, however, the MAC x_{k-i} of A_i should be received within time { S_t + (i-1) * T - D_s - D_r + delay}. If the MAC arrives late and is not received within this deadline, then there is a possibility for a MAC to be forged since the corresponding key could have been released by that time. As the key is released, anybody can generate a duplicate MAC as if MAC sent from a valid sender. If the MAC is received very late, they will be verified and marked as non-authentic and further blocks will be authenticated normally. These non-authentic MACs are called as holes and steps should be taken to minimize the number of holes in this scheme. These holes can be reduced to a great extent if the delay factor is increased and may lead to increase in

authentication latency. Thus the selection of small time frame to increase the latency on the downside may increase the number of holes created.

There are two major drawbacks in this scheme. One problem is to synchronize the receivers with the sender's clock as significant jitters may lead to delay in receiving the MAC resulting in holes. This is overcome by the forthcoming scheme [Perrig et al] where synchronization is improved to minimize the errors. Another issue is that the length of one way chain is bounded and requires new key chain to be replaced periodically. The periodic updates are also necessary to all the subscribed receivers which further increase the communication overhead as well as the need to use digital signature over the first MAC. [Groza07, Groza08] proposed a protocol based on time synchronization using one-way chains constructed with the squaring function that enables to construct a one-way chain of unbounded length. This permits the messages to multicast over long period of time without replacement of new one-way chain periodically and frequent announcements.

2.4.1 Time Asymmetry Basics

2.4.1.1 One way chain

In order to authenticate multicast messages, the sender and receiver should agree to use MAC keys and hence the sender needs to securely transfer the secret keys to the peer otherwise there is a possibility for any attacker to impersonate and send duplicate keys as if a valid sender. There is no feasibility to sign each secret key since the multicast requires fast transmission of keys to the receiver and any delay will interrupt them from binding to real time deadlines. One way key chain is the widely used cryptographic primitive to solve this problem. The main idea is to

certify the end node with a signature and recursive one way computation of the root node will lead us to the end node. As the end node is signed, the conclusion is made that all the intermediate nodes are certified and sent from the valid sender. The one way chain is illustrated in figure 2.4.1.

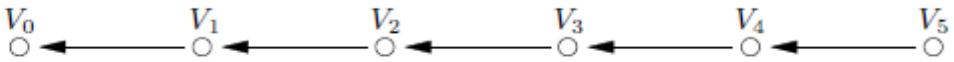

Figure 2.4.1.1: One way chain

There are two major properties for one-way chain generation and both completely depend on parameter one-wayness.

- The one-way function needs to provide the second pre-image collision resistance called as weak collision resistance which prevents another value V_i'+1 which satisfies condition V_i= H(V_i'+1) and V_i+1 ≠ V_i'+1.
- If the condition V_i= H(V_i'+1) and V_i+1 ≠V_i'+1 gets satisfied, then the attacker can forge the value V_i = H(V_i'+1) and will result in the verifier unable to authenticate future values such as V_i'+1 = H(V_i+2) = V_i+1 is probable.

Consider the collection of values (V_0... V_n) such that each value V_i except the last value V_n to be one way function of the next value V_i+1. The function which satisfies this condition is called as one way hash chain. i.e., we will have V_i = H (V_i+1), for 0 ≤ i < n and H is the one way cryptographic hash function.

At the sender side, the setup involves choosing a root of chain whose value is V_n and derives all previous values V_i from n-1 to 0 by recursively applying the hash function H. The value V_0 is referred as end value which is signed by the sender and sent to the receiver.

At the receiving end, the value of V_0 is the end value of generator's one-way function and its authenticity is proved with the signature appended by the valid sender. When any of the intermediate values such as V_i of a chain reaches the receiver, the hash function H is recursively used for i times to end up with value V_0. The value V_0 computed iteratively is compared with already stored value V_0 which is accompanied by the valid sender's signature. If both the value matches, then the value originated from the valid sender, else the value is ignored. Another advantage of one way function is that, in case another value V_k is received such that k < i and the value V_i is already authenticated by the receiver, the verifiers does not need to compute k one way function H to get the V_0 value. Instead, it can perform (i-k) times one way function to the input value and compare the V_i value with already verified V_i value sent from the sender. In applications such as TESLA, we require the generator to disclose successive values of one way chain and this scenario is called as one-way chain traversal.

The major advantage of the one-way chain is its secure one way function property. Consider, V_i as the latest verified key and V_j is expected to arrive from the valid sender. There is no possibility for the attacker to track or reverse engineer the V_j where $j > i$ using the one way chain. On the other way round, the validity of V_j can be computed and assessed easily by verifying that $H_{j-i}(V_j) = V_i$, where $j > i$. The main drawback is the number of recursive operations required to be performed in order to verify the validity of the key. For example, $j- i$ operations are required

to verify V_j provided V_i is known. The repeated disclosure of one way chain values increases the communication overhead for the packets from sender to the destination.

Further there is always a possibility for the commitment on the end value of key chain to be lost and in case of one-way key chain, the verifier has to wait for the next value of the primary chain to be disclosed to recover it. Therefore, Sandwich-chain is introduced which would be discussed in section 2.4.1.3.

There are two techniques of simple traversal. In one technique, the generator stores all the values of the one way chain in memory and storage of each value of a one-way chain with length n would result in a cost *O (n),* storage *O (n)* and traversal *O (1)* since all the values are stored and no re-computation is necessary. On the other hand, the generator recomputed each value from the root value and this involves a storage *O (1)* and traversal of *O (n)*. [CJ02] and [Sella03] came up with new techniques to optimize the traversal and storage and the results obtained using these techniques decreased the one way function traversal cost to *O(log(n))*, storage to *O(log(n))* and efficiency by $O(log(n))^2$.

2.4.1.2 Different construction of One-way chain

[LN03] proposed a scheme based on hierarchical multiple level one-way chain where each and every primary chain root acts as a seed for the secondary chains. The i^{th} value of the primary chain will contain the root of the i^{th} secondary chain and in this scheme all the values of the j^{th} secondary chain must be released along with the primary chain value V_j before the j+1 secondary chain is released.

The setup for this scheme involves picking V_n, which is the end node and compute the primary chain V_{n-1},...,V_0. Once the primary chain is decided, the secondary chain is precompiled on the fly and efficient scheme relies on requiring only N/k operations to compute V_0 compared to N in legacy one-way chain where k is the secondary chain length. The repeated hash operation traverses the whole chain V_{10}, V_{11} , V_{12}, V_{20}, V_{21}, V_{22},…., V_{N0}, V_{N1}, V_{N2} as shown in the figure 2.4.1.2.

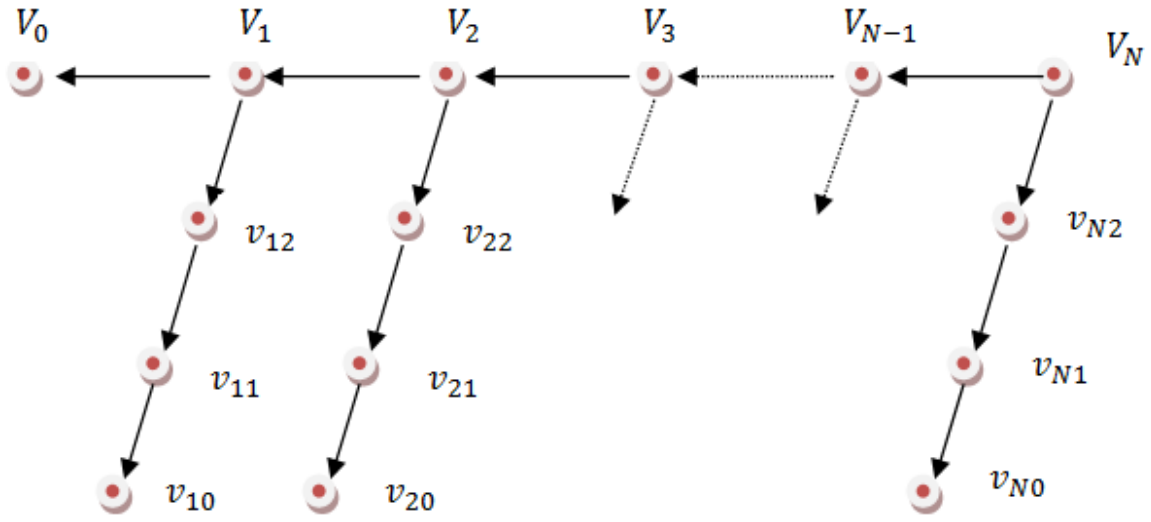

Figure 2.4.1.2: Hierarchical multiple level one-way chain

The major drawback is the loss of authentication message V_i will prevent the verification of V_{i1}, V_{i2}, V_{i3} until the next primary chain value V_{i+1} is known to the receiver.

2.4.1.3 The Sandwich-chain Construction

Similar to the hierarchical one-way chain, it consists of primary chain one-way-chain and each node in the primary one-way chain connected with a set of secondary one way chain. As the main issue with the hierarchical one-way chain is that in case there is a loss of end values of the secondary chain, then the verifier need to wait till the next primary chain value is known. To avoid this, we have third one-way chain W connected to the end values of the secondary chain. Even though, the transferred values are lost, they can be recovered using the one-way function W without waiting for the next primary chain value V_n [JP05].The sandwich-chain construction is illustrated in figure 2.4.1.3

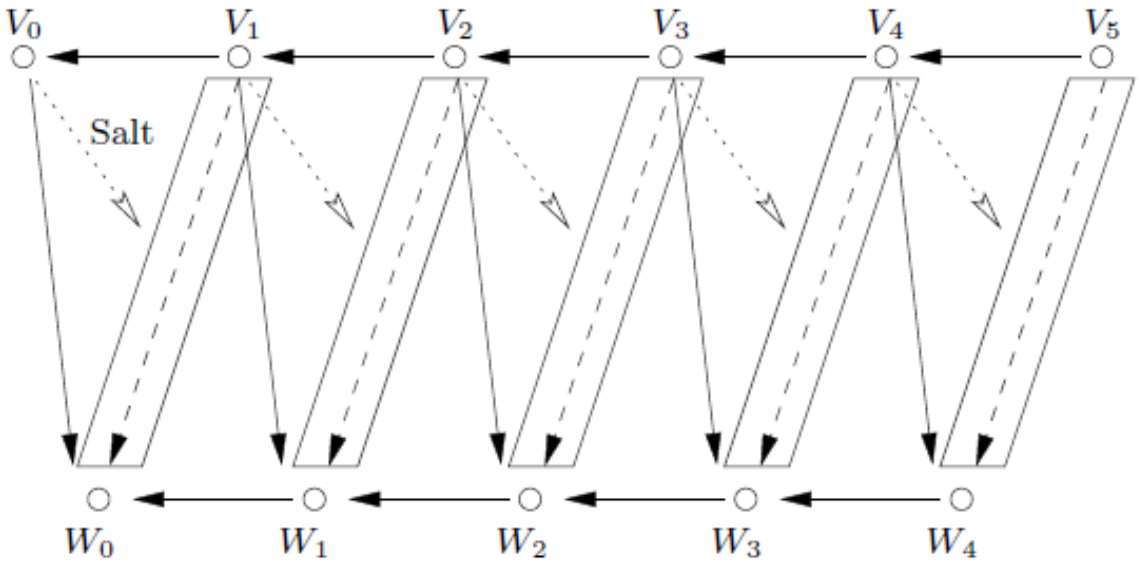

Figure 2.4.1.3 Sandwich-chain construction

2.5 Time Asymmetry protocols

2.5.1 BiBa

Adrian Perrig proposed a signature scheme that uses one-way function without trapdoors called BiBa (Bins and Balls signature). This is based on the principle based on collision of balls under a hash function in bins forms the signature [Perrig01].

BiBa scheme uses public validation information named SEALs (Self Authenticating values) which are random numbers generated for the receiver to authenticate the message using the public key instantaneously. The property of SEAL is that only the verifier will be able to authenticate the SEAL with the public key and it is computationally infeasible for an intruder to forge the message even though the public key is known, since the SEAL is unknown to the adversary. Moreover, the receiver is loosely time synchronized with the sender in Biba scheme.

Consider a sender throws the set of balls using a family of hash function G_h such that number of two-way collision into same bin forms the signature .This is under the assumption that the sender has many number of balls to throw and uses the set of hash functions G_h to compute the signatures. However, the sender sends only a partial list of balls from the subset used for signature such that the verifier uses the required number of balls using the hash function to verify the BiBa signature. At the same time, it is infeasible for the verifier to generate the signature as he lacks the complete information available with the sender. However, as the subset of keys is shared among many receivers, there is always a chance that a group of fraudulent receivers may forge by generating and impersonating as a valid sender signature. To avoid this issue, the sender changes the key modules periodically before the receivers know all the key subsets.

Experimental results show that the probability for the adversary to forge a valid signature is small compared to TESLA. This is illustrated in figure 2.5.1.

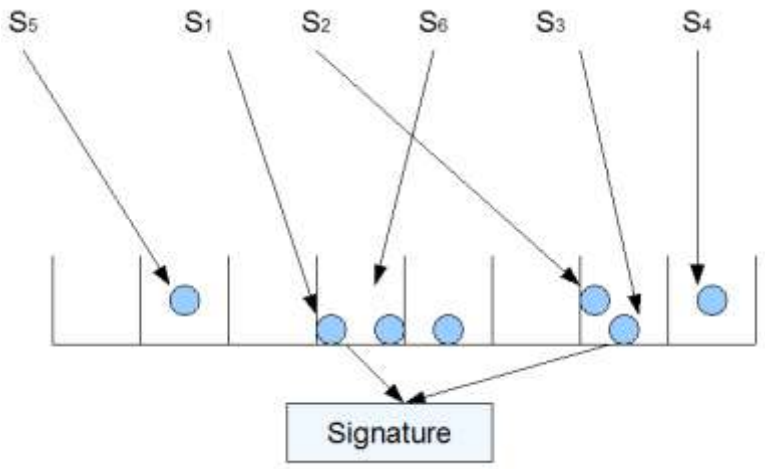

Figure 2.5.1: BiBa Scheme

The discussion regarding the BiBa broadcast authentication protocol design is follows. As mentioned earlier, the public key used to authenticate the messages is transmitted to the verifier in a secure way in advance and it is not feasible for the verifier to generate the message using the public key due to presence of only subset of keys. This scheme consists of t balls $\{S_1, S_2, S_3,\ldots,S_t\}$ which are random numbers considered as keys and n bins ranging from [0, n-1] are used as hash functions. The sender first computes the hash of the message m using the hash function, h = H (m | c) where c is the counter value. The counter gets incremented until the sender is able to find the valid signature. Then the signer uses the hash function G_h to sign all the balls $\{S_1,\ S_2\ , S_3, \ldots, S_t\}$ and looks for two-way collision such that $G_h(S_i) = G_h(S_j)$ and all the values of SEALs are distinct. The counter value gets incremented if the two-way collision is not found and the process is repeated till it finds a two-way collision. The pair $\{\ S_i, S_j\ \}$ forms the signature. When a message M along with the BiBa signature $\{\ S_i\ , S_j\ , c\ \}$ reaches the receiver,

the verifier computes h = H (m | c) and checks for distinct Balls and condition $G_h(S_i) = G_h(S_j)$, thus verifying the BiBa signature.

There are number of ways in which the security of the Biba can be improved.

- The straightforward way is to increase the number of SEALs and bins which in turn increase the public key size to great extent.
- To increase the number of two-way collisions required to sign a message with h = H (m) instead of one.
- To choose z-way collisions, instead of one two-way collision such that signature of the message m with h = H (m) is $(S_1, S_2, S_3 \ldots . S_t)$ where all SEALs are distinct and collide under $G_h(S_1) = G_h(S_2) \ldots . = G_h(S_t)$.
- To use multi-round scheme in which the bins having K_1 way collision in the first round proceed as balls into the second round.

When compared to other signature schemes, the major advantage of BiBa is the use of small signatures and low verification overhead to verify signatures. Also this scheme uses the one – way chain for key generation and provides perfect resistance to packet loss. However, the public key size is large and the signature generation overhead is very high compared to the other schemes.

2.5.2 Hash to obtain random subsets (HORS)

This scheme is designed to overcome the major drawback of BiBa which incurs long time to sign a message compared to other one-time signature schemes as discussed previously. This scheme offers better signing and verification time compared to BiBa and offers slightly improved key

and signature sizes. HORS scheme uses a hash function H to map each message M to a unique t-element subset of an N-element set, T .T forms the private key and public key is created by applying one-way function to each element T, and the t-element subset forms the signature [RR02].

Consider the private keys generated as S_k = ($s_1, s_2, s_3 \ldots, s_t$) with each containing randomly generated *l*-bit strings and public key P_k = ($v_1, v_2, v_3, \ldots, v_t$) = (f ($s_1$,)$f(s_2)$,$f(s_3)$,…,$f(s_t)$), where f is a one-way function. The sender signs each message m by computing h = H (m) where H is the hash function. Now split the h into k substrings ($h_1, h_2, h_3 \ldots, h_k$) of *log t* bits each. Where each h_j is interpreted as integer i_j and j varies from 1 ≤ j ≤ k, then the signature of m is given by ($s_{i1}, s_{i2}, s_{i3} \ldots, s_{ik}$).

To verify the signature ($s'_1, s'_2, s'_3, \ldots, s'_t$) over the message m, the verifier has to compute h = H(m). Now, the verifier will compute h = H(m) and split h into k substrings h_1, h_2 , $h_3, \ldots, h_k$ of *log t* bits each. When each h_j is interpreted as integer i_j, the signature is verified by computing f (s'_j) = v_{ij} for each j 1 ≤ j ≤ k, otherwise rejected.

Though the scheme has fast signing and verification time, the public key size of HORS and the signature key size are in order of O(N) which is costly compared to all other schemes. The public key size can be reduced using the design proposed by Chang et al [CSL+06].

2.5.3 Powerball

Mitzenmacher and Perrig [MP02] proposed an enhanced scheme to improve the signature and verification time of BiBa.In Biba scheme, there are fixed number of known signature patterns

such that a collision of k balls in one bin forms a signature pattern. In Biba, all patterns agree on that and the scheme is implicit. Powerball is explicit in the sense, the signer commits to t balls in the public key and also commits to t' patterns P_i (1< t < t'). Each pattern denoted k bins such that P_i = (b_1,……, b_k).

In Powerball scheme, the sender computes a hash of the message m such that h = H (m | c) where c is the counter which is incremented when a valid signature is not matched and h is used to select a one-way function G_h from the family of function G. G_h maps each ball to one of the n bins and the signer searches for a pattern P_i such that every bin in the pattern contains the ball and creates the signature ($B_{\alpha 1}$,……, $B_{\alpha k}$, P_i,c) with α_j denoting the indices of balls that landed in the bins of pattern P_i.

The verifier performs the below steps to authenticate the signature ($B_{\alpha 1}$,……, $B_{\alpha k}$, P_i,c) on message m,

- Check for $B_{\alpha i} \neq B_{\alpha j}$ for i ≠ j so that signature are unique.
- Check for the authenticity of balls F ($B_{\alpha i}$) using the public key and verify the same.
- Check for the authenticity of the pattern F(P_i) using the public key
- Check h = H (m |c) and check G_h from the one-way function family and verify k balls. covering all k bins of pattern $P_i = (b_1, b_2, \ldots, b_k)$ such that $G_h(B_{\alpha 1}) = b_1$, $G_h(B_{\alpha 2}) = b_2$, …. , $G_h(B_{\alpha k}) = b_k$.

This scheme has high susceptibility to DoS attacks because even if forger can find a signature where k-1 balls land in the correct bin, a verifier that checks the balls of the signature discovers the bad ball after checking an average of (k+1)/2 balls. This way even if the forger somehow find

fewer matching balls, the verifier can still detect the invalid signature after few hash function computations. However, on the downside, these improvements come at the cost of larger public key which is twice as large as the BiBa signature scheme.

2.5.4 TESLA and related protocols

Timed Efficient Stream Loss-tolerant Authentication (TESLA) is an improved version of the scheme proposed by Bergadano et al [BCC00, BCC2000]. The major issue with the idea proposed by Bergadano et al is the synchronization issues between the sender and the receiver which results in holes. To overcome this, Perrig et al came up with TESLA which uses loose time synchronization between the sender and the receiver. It incurs low communication and computation overhead and also scales to large number of receivers tolerating packet loss. The main idea of TESLA is that the sender appends a MAC to each packet which is computed with a key that is initially known only to itself and dispatched the packets to the receivers. When the packets are received, they are immediately buffered as the receiver has no knowledge of the key to verify the message. The keys are disclosed after a delay and the receiver is able to authenticate the message. As discussed, the receiver clock is loosely synchronized with the sender and computes a bound within which it expects the key to reach the receiver to authenticate the message. If it receives the keys for a specific message beyond the time deadline, it assumes the packets are tampered by the intruder and ignores the message. As,TESLA uses one way key chain, when the receiver receives future keys from the sender, it authenticates the packets which are pending to be authenticated due to late arrival of keys or key lost in the lossy network.

2.5.4.1 Time synchronization mechanism with TESLA

TESLA uses loose time synchronization between the sender and receiver which is the real time δ taken for a message to reach the receiver from the sender. In loose time synchronization, the receiver knows the upper bound on the sender's local time and computes Δ which is the maximum time synchronization error. A setup phase of simple time synchronization mechanism used in TESLA to compute the Δ is shown in figure 2.5.4.1.

The sender and receiver share a digital signature to authenticate the messages from both nodes. First receiver sends a time synchronization request at time t_R at which the sender time is t_1 with randomly generated nonce. When the message reaches the sender, the message is authenticated with the public key. If the message is authentic, then the destination stores t_R and t_s and calculates an upper bound on the sender's clock time at local time t as $t - t_R + t_S$. It also creates a response packet containing the sender time t_S and Nonce and signs with the private key which is sent to the receiver.

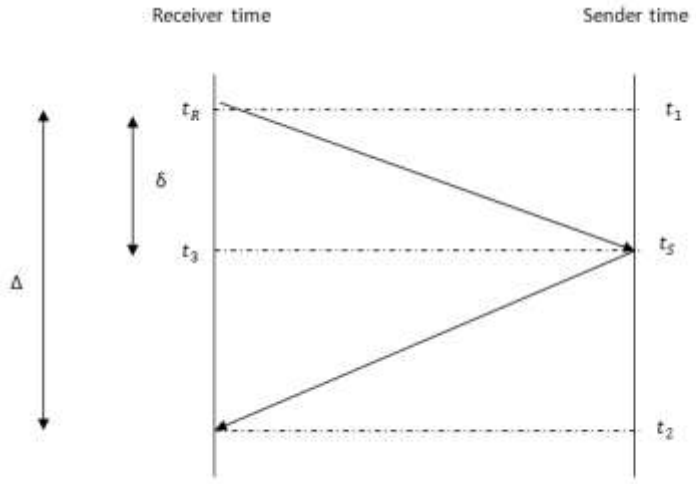

Figure 2.5.4.1**:** TESLA setup phase

When the receiver gets the message, it validates the signature using its public key and compares the value of random nonce in packet with the generated one at the receiver during time t_R. After successful authentication, the receiver will have current time t_r along with t_R and t_S appended in the received message packet. Now the receiver computes the upper bound on the current sender's time which is $t_s \leq t_r - t_R + t_S$.

2.5.4.2 TESLA broadcast authentication protocol design

TESLA uses the time of asymmetry such that the receivers can only verify the authentication information and at the same time, the generation of valid authentication information is infeasible due to assumption that the receivers are loosely time synchronized with a synchronization error Δ and the end nodes agree on this during the setup phase.

The sender splits the time into equal duration interval and uses one key per time interval generated by the one-way chain of self-authenticating values. At the sender side, MAC is computed for each message and the MAC key used for the corresponding message is generated using the one-way function. The sender appends the MAC and the most recent one-way key chain value with the message which is sent to the receiver.

At the receiver, each incoming message packet is buffered and checks for the disclosing time interval of the received message. The disclosed keys in the current packet are used to validate the already received packets in the buffer and the receiver accepts the packet if MAC is correct. The salient feature of TESLA is the use of one-way key chain and its capability to recover lost keys to authenticate the buffered packets incase the corresponding keys to

authenticate the previous message were lost. We have discussed about one-way key chain and how we derive the lost keys in Section 2.4.1 about one-way key chain.

In the figure 2.5.4.2, consider one-way key chain using the function F and the derived MAC keys using one-way function F'. The time is split into uniform duration periods and increases from left to right. P_j, P_{j+1}…,P_{j+x} are the packets which contains key that is disclosed to compute the MAC of the packet.

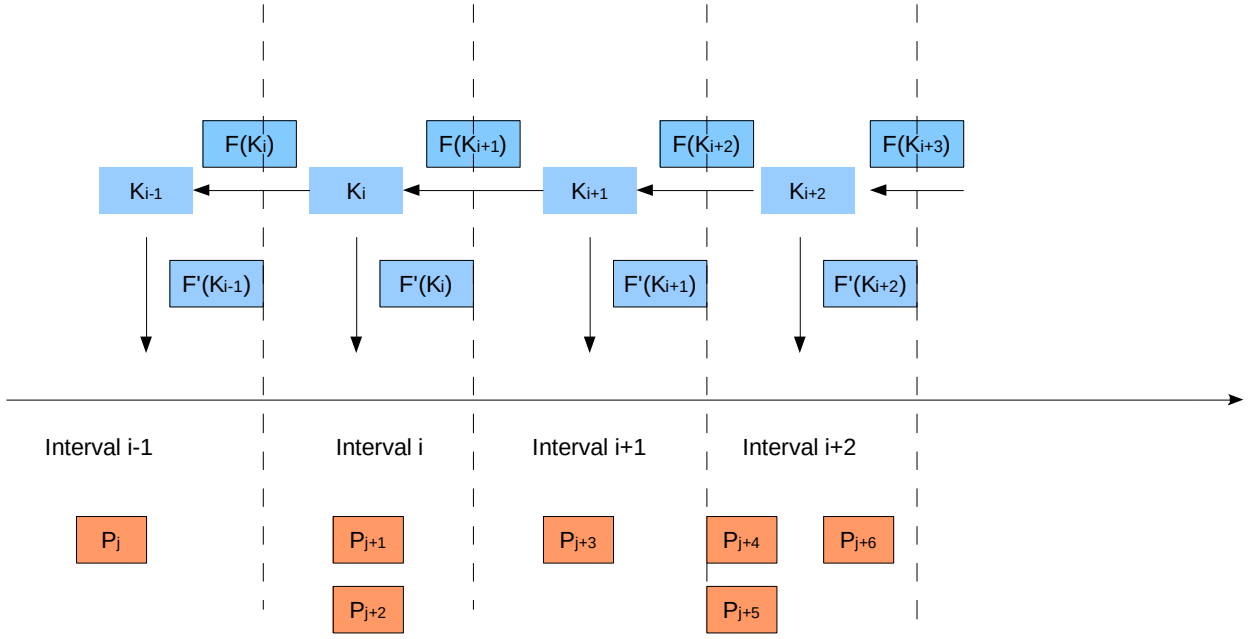

Figure 2.5.4.2: TESLA protocol architecture

TESLA has some drawbacks where the delay is incurred at the receiver due to buffering and verification.The receiver has to buffer the packets until the corresponding key k from the sender is received. This increases the end-to-end latency and delay the delivery of information to the corresponding application. This also adds extra overhead as it requires the extra storage space and increases the vulnerability to DoS attacks on the receiver end since bogus packets can be

flooded. One solution to this issue is to buffer the packets at the sender side instead of buffering at receiver end. This way we can reduce the end-to-end latency as well as vulnerability to DoS. When we buffer the packets at the sender end, in turn we increase the disclosure delay by one cycle. Hence, when packet $P1$ is buffered at the sender side for a disclosure, the delay packet P_2 will be ready to be processed at that instant. Furthermore, the hash value of the packet P_2 is computed and piggybacks it to packet P_1. When the packet P_2 reaches the destination, there is no necessity to wait for its hash key to be sent in future packets since the hash computed and piggybacked to packet P_1 can authenticate packet P_2 instantly. This is called as immediate authentication for TESLA. To achieve flexibility and robustness to packet loss, multiple hash values of future packets can be sent in a packet.

In TESLA, the sender has to authenticate time synchronization with each receiver individually. This will not scale well for n number of receivers or in a multicast environment where there will be many receivers trying to synchronize with the sender since synchronization requires public key operation which is computationally expensive. [ARD+01] paper proposes the use of single public key operation for multiple time synchronization setup in TESLA. [ARD+01] also discusses how to choose the key disclosure delay for direct and indirect time synchronization which is a crucial parameter for TESLA.

Another assumption with TESLA is that all the receivers are already subscribed to the sender and synchronized with the sender before transmission is initiated. This kind of situation is impossible and there are many instances where the receivers join to multicast group at later point of time and get synchronized. [ARD+01] discusses an idea on how the receivers can “join in a fly” with the ongoing session and get synchronized at the later time.

In TESLA, the verification delay is normally determined by the maximum time taken by a multicast packet to reach all the nodes in the network. This parameter verification delay determines how long the packet needs to be buffered at the receiving node before its authenticated. In large networks, the verification delay may be high depending on the size of the network as the delay for all the nodes increases in accordance to the maximum verification delay. This verification delay can be reduced by using L-TESLA. This scheme considers few nodes in the network as trusted nodes and divides into several subsets and coordinate multicast authentication within the subsets. The trusted nodes coordinate multicast authentication in each subnet so that the nodes in subnet can verify the message as if they are in small network [DG06].

2.6 Time Valid-One Time signature

The major challenge in any multicast authentication system is to provide time critical authentication to incoming packets as well as to avoid buffering of packets so that data can be processed immediately. In any multicast authentication system such as power grid, small communication overhead, packet loss tolerance and resistance of the network against malicious attacks are desirable. To boost the efficiency of regular one time signature scheme, this paper proposes a new signature model named “Time Valid One Time Signature” (TV-OTS) [Wang09].

One-way hash chain is combined with TV-OTS to sign large number of streaming packets for which efficient multicast authentication scheme named “Time Valid Hash to Obtain Random Subsets” (TV-HORS) is designed. It provides fast signing/verification as well as packet loss tolerant. It is also robust to malicious packets and provides buffer free data processing which reduces end-to-end latency making it one of the fastest authentication schemes. The communication overhead of TV-HORS is smaller than regular OTS schemes and RSA signature.

One-time signature has the advantages of fast message signing/verification and buffer free data processing. However, the disadvantage of one time signature is the large signature size which increases the end-to-end latency [Per01]. This can be solved by using smaller signature size in TV-OTS which uses only first L bits of the hash of the message to sign thereby increasing the efficiency to authenticate large volume of messages. For example, if the original message to be signed is 160 bits of the hash. By this new scheme, we improve the efficiency of OTS scheme by signing first 40 bits of the message digest instead of complete hash.

The only way for an intruder to know the partial hash collision is brute force and the number of hash computations needed to find L bits is 2^L on an average. Thus, thousands of second are required to find the signature using brute force attack. To prevent signature forging, TV-OTS uses a loose time synchronisation in which receiver can identify an upper bound on the synchronization error with the sender. In this model, the sender signs the message and specifies a signature period in advance within which the message has to reach the receiver. Suppose, if the message reaches receiver within the specific time bound, receiver is assured that the intruder has not obtained the valid signature and the receiver can continue with the verification otherwise receiver will discard the message.

Let t_0^s be the local starting time of the signature period, t_1^s be the local ending time of the signature period, σ is the maximum end-to-end delay and t_{int} is the calculation time for the intruder to find the possible second pre-image partial collision. The signature period is selected by the sender and the time period within which the receiver has to obtain the message is embedded into sender's public key.

Whenever a signed message is received at the receiver R, the local receiving time t^R is initially recorded and the upper bound on the S's local time is estimated as $t^R + \epsilon$ where ϵ is the synchronization error with respect to the sender. Once the sender signs the message at the beginning of the signature period, the longest exposure time of the signature from the time sender signs the message and it is received at the destination given by $t^R + \epsilon - t_0^s$. When the condition $t^R + \epsilon - t_0^s < t_{int}$ is satisfied, the receiver is assured that the intruder is unable to find any matching partial collision and continue to verify the signature. On the other hand, if $t^R + \epsilon - t_0^s > t_{int}$, the receiver stops verifying the signature as it assumes that there is a possibility for the intruder to modify or inject malicious content into the original message sent by the sender [Wang09]. This is illustrated in figure 2.6.

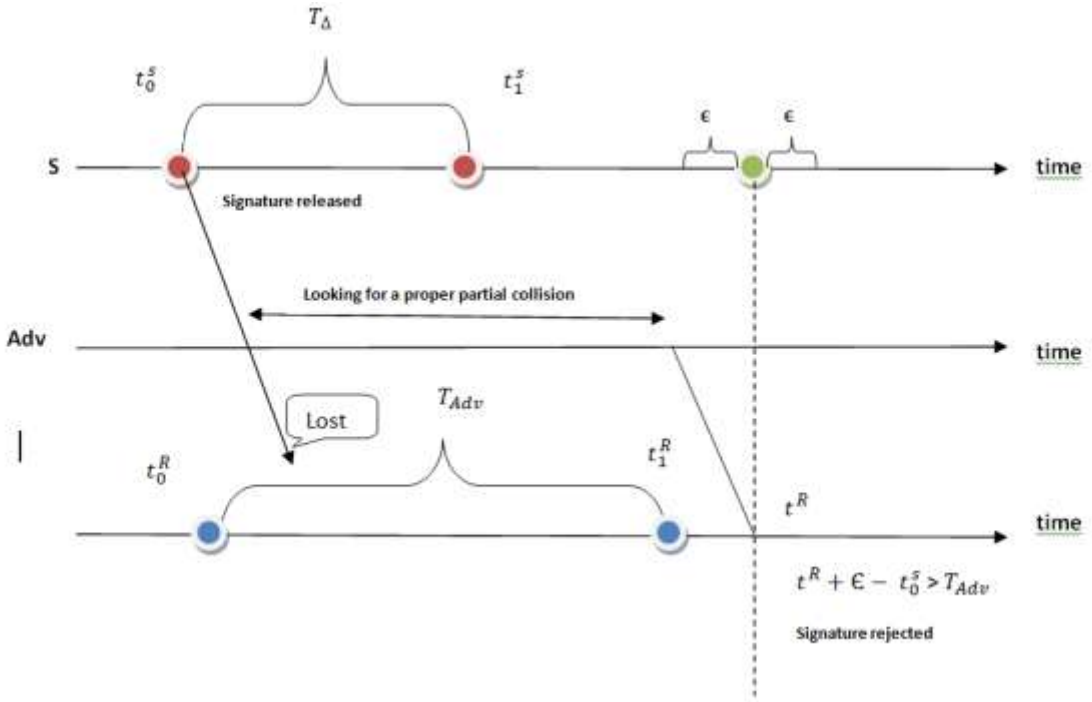

Figure 2.6: TV-OTS scheme

Hence, TV-OTS is designed to provide required security with low computational cost since it signs only part of the message digest rather than signing the whole hash of the message as in standard OTS schemes. It also uses loose time synchronization between the communicating peers. Moreover, the signature contains the upper bound timestamp within which is expected to be verified at the receiver.

Using this scheme, we are able to achieve very fast signing and verification where signing involves one hash computation with low end-to-end computational delay in the range of microseconds. Moreover, TV-OTS just requires loose time synchronization between sender and the receiver. However, the drawback is generating key pairs in computationally expensive and the use of large public key of size 8Kb to 10Kb [Wang09].

CHAPTER 3

Evaluation and Results of different types of cryptographic protocols

The actual latency and computation time of multicast authentication schemes have to be assessed for the use in power grid applications. Table 3.1 expresses the computation time, message overhead, and buffering requirements associated with the various protocols in terms of more primitive operations such as cryptographic hashing or symmetric cryptography.

Symmetric key cryptography is a simple authentication mechanism which involves the use of single key to encrypt at the publisher side and the same key is used to decrypt the message at the receiving end. The total latency involves the encryption, decryption and network delay. The major drawback is the use of symmetric key which fails to provide non-repudiation which is often considered to be one of the important requirements of multicast authentication schemes. However, this can be overcome by the use of public key cryptography which utilizes a private key to sign the message at the sender end and public key to verify the message. This provides non-repudiation, on the other hand the computational cost is too high which increases the total latency incurred from publisher to subscriber.

The individual authenticity can be provided by time synchronized protocols such as TESLA which uses a specific bounded time delay in disclosing the keys for the message that has been sent at specific time interval. In these schemes, a single hash operation is involved at the sender where the computed hash of the message is appended to the end of the message. However, at the receiving end, the hash is split from the message and hash operation is

performed on the message using the key at the receiver end. Authenticity is verified by comparing this computed digest with the hash with the appended message. The additional parameter called key disclosure delay is involved since the keys are disclosed after some specific timeline. These keys received after an interval will be used by the receiver to authenticate the messages received in calculated timeframe.

Scheme	Sender comp. cost	Receiver comp. cost	Communication overhead	Packet buffering		Non-repudiation	key size	Total latency
				Sender	Receiver			
AES	1E	1D	1k	1	1	No	O(1)	1E+1D+ND
RSA & DSA	1S	1V	1k	1	1	Yes	O(1)	1S+1V+ND
TESLA	1H	1H	1k+1h	1	Є	No	O(1)	2H+ND + kD
TV-OTS	1H	1 H	0.25*h	1	1	No	O(N)	2H+ND
Tree Based	TH+ (S/n)	log nH+ (V/n)	s + log nh	n	1	Yes	O(1)	(T+ log n)H +(S+V/n)
Augmented chain	1H+ (S/n)	1H+(V/n)	2h+ (s/n)	p	n	Yes	O(1)	2H+ (S/n)+(V/n)
EMSS	1H+(S/n)	1H+(V/n)	2d+ (s/n)	1	n	Yes	O(1)	2H+ (S/n)+(V/n)
Erasure code	1H+(S/n)+ EC	1H+(V/n)+ ED	(s/x)+(n/y)h	x	y	Yes	O(1)	2H+ (S/n)+(V/n)+EC+ED

E- Encryption D-Decryption Є - number of packers per key disclosure delay N - Number of hash chains used h - Hash size S – Signature

KD - Key disclosure delay k- no of keys H - Hash function M - Message authentication code ED - Erasure decoding EC - Erasure coding

t - Number of SAGES's contained in each signature V – Verification m - denotes the number of siblings from the path along the leaf node to the root

n - Block size (x,y) denoted the splitting the message into x pieces where y < x pieces are sufficient to retrieve the message where x-y is considered as lost

D - Average distance between the current SEAL and the last verifies SEAL in the same SEAL chain TESLA Timed Efficient Stream Loss-Tolerant Authentication

Table 3.1: Theoretical Performance of Authentication Protocols

The latency increases with the increase in key disclosure delay to a great extent and the designer is responsible to come up with a reasonable key disclosure delay. The major advantage of using

time synchronized protocols is its ability to achieve asymmetry through clock synchronization and delayed key disclosure.

The three variants such as public key cryptography, symmetric key cryptography and time synchronized protocols are discussed previously with their theoretical performance in scenarios. Further, improved schemes are implemented using signature and MAC authentication in order to provide reasonable performance. Tree based authentication schemes discussed in section 2.3.2.1 are individually verifiable and can tolerate packet loss. However, the communication overhead is considered to be very large since each packet must carry signature of the root and hashes of the sibling of packet from the current path to the root to authenticate the message. Moreover, the packets which wait for authentication information need buffering at the receiver which increases the end-end latency.

The major drawback of tree based chaining is the use of signature in every packet which increases the latency. This overhead can be reduced by using EMSS in which a single signature is used to span multiple packets. However, loss of single packet will break the chain and make it impossible to verify the authenticity of packets preceding the break point. The packet loss tolerance can be improved further by using augmented chain techniques. In this scheme, the hashes of packets are included at strategic locations in a deterministic way. Hence, the chain of packets formed by the hashes will be optimally resistant to bursty packet loss. However, this scheme does not provide complete fix to packet loss issue.

An alternative way to address the packet loss issue is to use forward error correction techniques. This technique splits the message into m pieces such that the reproduction of the original message is possible even if n-m packets are lost. This provides reasonable robustness to

packet loss with low computation overhead. All the four special schemes mentioned above have three major factors known as verification probability, communication overhead and latency. The use of a specific protocol completely depends on the specific application requirement and the particular drawbacks of the protocol should have minimal effect on the application where it is used.

Table 3.2 discusses the measurements of computation time associated with various authentication techniques. The computation time at the publisher and subscriber are very critical since they are major factor in deciding the end-end latency incurred.

Algorithm/protocol	Publisher Computation cost (ms)	Subscriber Computation cost (ms)	Total computation cost (ms)
AES (128 bit)	**0.04**	**0.03**	**0.07**
RSA (2048 bit)	**59.00**	**2.04**	**61.04**
DSA (1024-bit)	**5.10**	**9.80**	**14.90**
TV-OTS	**0.04**	**1.42**	**1.46**
TESLA	**0.03**	**0.03**	**0.06**
HMAC-SHA1	**0.02**	**0.02**	**0.04**
CMAC	**0.04**	**0.04**	**0.08**
SHA-256	**0.01**	**0.01**	**0.02**

Table 3.2: Computation cost for different algorithms at Publisher and Subscriber nodes

The above results are obtained using cryptographic functions from the Java SE Runtime Environment running on the Java Hotspot 64-bit server VM from Java version 1.6.0_26 , an AMD Phenom II X4 920 processor with a 2.8Ghz clock, and Ubuntu Linux version 10.10. These

authentication schemes provide us with a general idea on the usage of different authentication keys depending on the application. Figure 3.1 shows the publisher and subscriber side computation cost for different multicast authentication protocols. The average network delay is assumed as 15ms and the key disclosure delay for TESLA is assumed as 25ms.

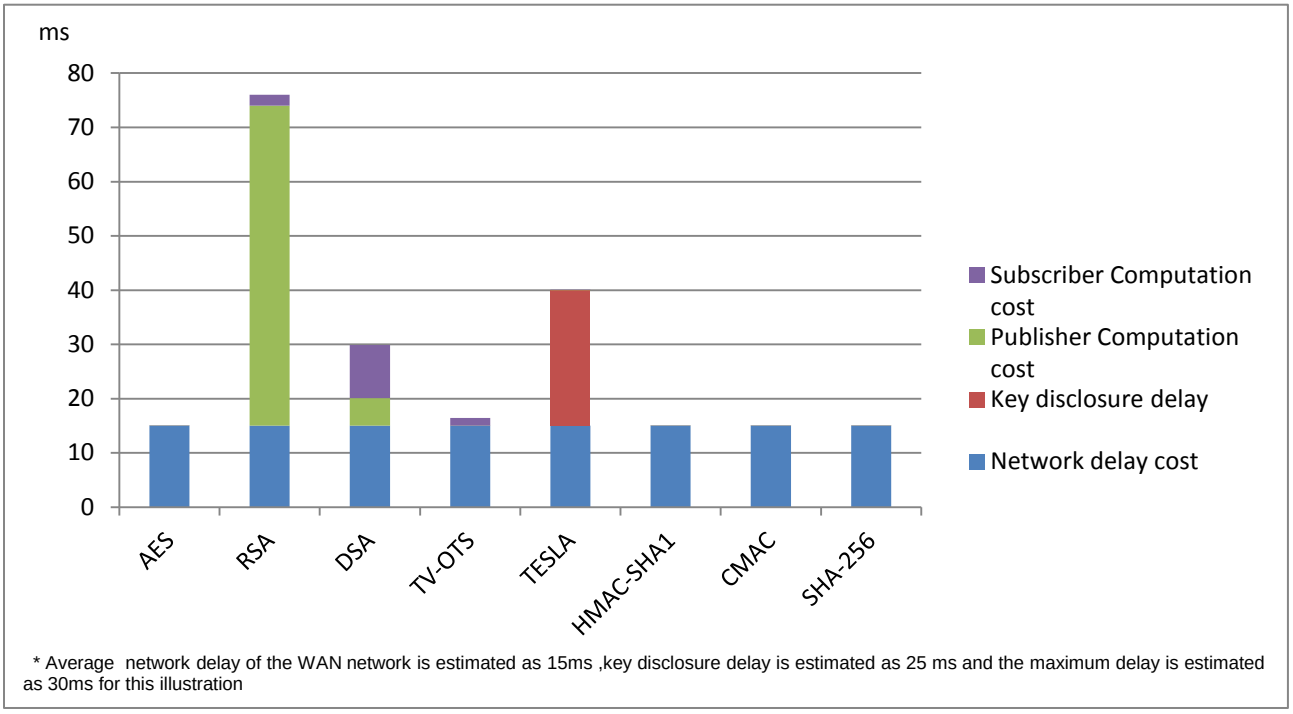

Figure 3.1: Performance results of different multicast authentication algorithms

CHAPTER 4

Conclusion and Future work

Latency and security needs are inversely proportional to each other. The higher the effort to enhance the security of the critical infrastructure will increase the computational cost which in turn increases the latency. The experimental results show that the use of public key cryptography such as RSA and DSA has improved security providing non-repudiation. However, the computation cost is considered to be high due to mathematical exponential operations in public key cryptography. This increases the end-end latency from the publisher to subscriber thereby resulting in designers unable to use public key cryptography in time critical infrastructures. On the other hand, the use of symmetric key such as AES are very fast but are susceptible to key disclosure by single recipient leading to the possibility of forged messages. In addition, the use of asymmetric key provides non-repudiation which is sometimes a requirement in multicast message authentication.

We also analyzed the use of timed key released protocols like TESLA and TV-OTS for use of multicast message authentication in critical infrastructures. TESLA protocol use message authentication codes which are computationally inexpensive. However, TESLA requires good clock synchronization between the sender and the receiver. The disruption of clock synchronization will potentially have greater effects on the value of measurements themselves

and inability to authenticate the measurements in time. Furthermore, the keys to authenticate the packet are released after a synchronized delay between the sender and the receiver which requires packet buffering at the receiver. This fixed parameter key disclosure delay added to the end-to-end latency increases the message reception at the receiver by a great factor. The use of TV-OTS has both low latency and low per-message computation cost. On the other hand, large number of one time keys are required that are generated periodically. The use of Message authentication codes such as HMAC and CMAC are also discussed to provide authentication which are computationally inexpensive.

The results show that the critical infrastructures such as power grid applications require provision to accommodate different message authentication protocols for different applications and data origin. In accordance to the specific requirement of applications such as security need and acceptable latency, the designers are expected to decide upon the authentication protocol that can be used. There is no hard and fast rule that one specific protocol fits for all applications providing low latency with secured usage. Each protocol has its pros and cons and a decision has to be made in order to fit the advantages to the specific application requirements.

References

[ABC+98] Ross. J.Anderson, Francesco Bergadano, Bruno Crispo, Jong-Hyeon Lee, Charalampos Manifavas, and Roger M. Needham, "A new family of authentication protocols." *Operating Systems Review*, 1998.

[ARD+01] A. Perrig and R. Canetti and D. Song, and J.D. Tygar, "Efficient and Secure Source Authentication for Multicast," *In proceedings of Network & Distributed System Security (NDSS'01)*, 2001.

[Bal95] A. J. Ballardie, "A New Approach to Multicast Communication in a Datagram Network", Ph.D. Thesis, University College London, 1995.

[BBH+11] D. Bakken, A. Bose, C. Hauser, D. Whitehead, and G. Zweigle. "Smart Generation and Transmission with Coherent, Real-Time Data" *Proceedings of the IEEE (Special Issue on Smart Grids)*, June 2011.

[BCC00] Bergadano, F., Cavagnino, D and Crispo, B. "Individual single source authentication on the mbone," *In Proceedings of International Conference on Multimedia and Expo 2000*, 2000.

[BCC2000] F. Bergadano, D. Cavagnino, and B. Crispo, "Chained stream authentication," *In Proceedings of selected areas in Cryptography*, 2000.

[CGI+99] R. Canetti, R. Garay, J. Itkis, G. Micciancio, D. Naor, and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," *in proceeding of IEEE INFOCOM'99*, 1999.

[CBB+04] Y. Challal, H. Bettahar, A. Bouabdallah, "A taxonomy of multicast data origin authentication: Issues and solutions, " *Communications Surveys & Tutorials, IEEE*, vol. 6, no. 3, pp. 34-57, 2004.

[Challal et al] Y. Challal, H. Bettahar, and A. Bouabdallah, "A2Cast: an adaptive source authentication protocol for multicast streams," *IEEE-ISCC*, 2004.

[CJ02] D. Coppersmith and M. Jakobsson, "Almost optimal hash sequence traversal," *In Proceedings of the Fourth Conference on Financial Cryptography (FC '02)*, 2002.

[CSL+06] Shang-Ming Chang, Shiuhpyng Shieh, Warren W. Lin, and Chih-Ming Hsieh, "An efficient broadcast authentication scheme in wireless sensor networks," *ASIACCS '06 Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, 2006.

[DFY92] Y. Desmedt, Y. Frankel, M. Yung, "Multi-receiver/multi-sender network security: efficient authenticated multicast/feedback," *INFOCOM '92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE*, 1992.

[DG06] Jawad Drissi, Qijun Gu, "Localized Broadcast Authentication in Large Sensor Networks," *International conference on Networking and Services (ICNS'06)*, 2006.

[DH76] W. Diffie, M. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, 1976.

[FKK96] H. Fujii, W. Kachen, and K. Kurosawa, "Combinatorial Bounds and Design of Broadcast Authentication," *The institute of Electronics, Information and Communication Engineers Trans*, vol. 79 (1996), pp. 502-506, 1996.

[HHN09] Ying Huang; Wenbo He; K. Nahrstedt, "ChainFarm: A novel authentication protocol for high-rate any source probabilistic broadcast," *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, Oct. 2009.

[HJP05] Yih-chun Hu, Markus Jakobsson, and Adrian Perrig, "Efficient constructions for one-way hash chains," *In Applied Cryptography and Network Security (ACNS 05)*, 2005.

[JEH02] Jung Min Park, Edwin K. P. Chong, Howard Jay Siegel, "Efficient Multicast Packet Authentication Using Signature Amortization," *IEEE Symposium on Security and Privacy*, 2002.

[GM01] P. Golle and N. Modadugu, "Authenticating streamed data in the presence of random packet loss," *Network and Distributed System Security Symposium (NDSS '01)*, 2001.

[GR01] R. Gennaro and P. Rohatgi, "How to Sign Digital Streams," *Information and Computation*, vol. 165, no. 1, pp. 100–16, 2001.

[GR97] R. Gennaro and P. Rohatgi, "How to Sign Digital Streams," *Advances in Cryptology, CRYPTO'97*, 1997.

[GM01] P. Golle, N. Modadugu, "Authenticating streamed data in the presence of random packet loss," *NDSS'01 The Network and Distributed System Security Symposium*, 2001.

[Groza07] Bogdan Groza, "Broadcast Authentication Protocol with Time Synchronization and Quadratic Residues Chain," *The Second International Conference on Availability, Reliability and Security (ARES'07)*, 2007.

[Groza08] Bogdan Groza, "Broadcast Authentication with Practically Unbounded One-way Chains," *Journal of Software*, vol. 3, pp. 11-20, 2008.

[HBC06] Y. Hinard, H. Bettahar, Y. Challal, "Layered multicast data origin authentication and non-repudiation over lossy networks." *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC '06)*, 2006.

[HBK09] Hasan, R. Bobba, H. Khurana, "Analyzing NASPInet data flows," *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, Mar. 2009.

[HM10] M. Hefeeda, and K. Mokhtarian, "Authentication schemes for multimedia streams: Quantitative analysis and comparison," *ACM Trans. Multimedia Computer Communication*, 2010.

[HM97] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture", *RFC 2094*, Jul. 1997.

[HPJ03] Yih-Chun Hu, Adrian Perrig and David B. Johnson, "Efficient security mechanisms for routing protocols," *In Network and Distributed System Security Symposium, NDSS '03*, 2003.

[JGX+07] He Jin-xin, Xu Gao-chao, Fu Xiao-dong, Zhou Zhi-guo, "A Hybrid and Efficient Scheme of Multicast Source Authentication" *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing IEEE*, 2007.

[JP05] Y. Hu, M. Jakobsson, and A. Perrig, "Efficient Constructions for One-Way Hash Chains," *Proceedings of Network & Distributed System Security (NDSS'05)*, 2005.

[KO97] K. Kurosawa and S. Obana, "Characterization of (k, n) Multi-receiver Authentication," *Information Security and Privacy, ACISP'97,* LNCS 1270, pp. 205–215, 1997.

[Lam81] L. Lamport. "Password authentication with insecure communication*," Communications of the ACM*, 1981.

[LMS+97] M. Luby, M. Mitzenmacher, M. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," *ACM Symposium on Theory of Computing*, pp. 150– 159, 1997.

[LN04] D. Liu, and P. Ning, "Multi-level uTESLA: a broadcast authentication system for distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 4, pp. 800–836, 2004.

[LN03] D. Liu and P. Ning, "Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks," *In Proc of NDSS'03*, Feb.2003.

[OK01] S. Obana and K. Kurosawa, "Bounds and Combinatorial Structure of (k, n) Multi-receiver A-codes," *Designs, Codes and Cryptography*, vol. 22, no. 1, pp. 47–63, 2001.

[Merkle80] R. Merkle, "Protocols for Public Key Cryptosystems," *Proc. IEEE Symp. Security and Privacy*, Apr. 1980

[MI97] S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting", *Proc.of the ACM SIGCOMM '97, Cannes, France*, 1997.

[MP02] M. Mitzenmacher, and A. Perrig, "Bounds and Improvements for BiBa Signature Schemes," Technical Report (TR-02- 02), Harvard University, 2002.

[MS98] D. McGrew and A. T. Sherman, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees", 1998.

[PAW08] Peng Ning, An Liu, and Wenliang Du, "Mitigating DoS attacks against broadcast authentication in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, 2008.

[PCS+05] A. Perrig, D. Song, R. Canetti, J.D. Tygar, B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", *Internet RFC 4082*, RFC Editor, June 2005

[PC04] Y. Park, and Y. Cho, " The eSAIDA stream authentication scheme," *In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA'04),* Lecture Notes in Computer Science, vol. 3046. Springer, 799–807, 2004.

[PCS03] J. Park, E. Chong, and H. Seigel, "Efficient multicast stream authentication using erasure codes," *ACM Transactions on information and System Security*, 2003.

[PCS+00] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *IEEE Symposium on Security and Privacy*, 2000.

[PCW+01] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, "SPINS: security protocols for sensor networks," *in ACM MobiCom*, 2001.

[PCT+02] A. Perrig, R. Canetti, D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *Cryptobytes,* vol. 5, no. 2, pp. 2–13, 2002.

[Perrig01] A. Perrig, "The BiBa One-time Signature and Broadcast Authentication Protocol," *8th ACM Conference of Computer and Communication Security*, 2001.

[PM03] A. Pannetrat, and R. Molva, "Efficient multicast packet authentication," *In Proceedings of the Network and Distributed Systems Security Symposium (NDSS'03), 2003.*

[PM02] A. Pannetrat, and R. Molva, "Authenticating Real-Time Packet Streams and Multicasts," *7th Int'l. Symp. Comp. and Commun., ISCC'02*, Jul. 2002.

[QKY+09] Qiyan Wang, H. Khurana, Ying Huang, K. Nahrstedt, "Time Valid One-Time Signature for Time-Critical Multicast Data Authentication," *INFOCOM 2009, IEEE*, Apr. 2009.

[Rab89] M. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *Journal of the ACM,* Vol. 36, No. 2, pp. 335–348, 1989.

[RR02] L. Reyzin and N. Reyzin, "Better than BiBa: Short One-time Signatures with Fast Signing and Verifying," *7th Australian Conf. Info. Security and Privacy*, 2002.

[RSA78] R. L. Rivest, A. Shamir and L.M. Adelman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, pp. 120–126, 1978.

[SJ01] Sara Miner, Jessica Staddon, "Graph-Based Authentication of Digital Streams," *IEEE Symposium on Security and Privacy*, 2001.

[Sella03] Yaron Sella "On the computation-storage trade-offs of hash chain traversal," *In Proceedings of Financial Cryptography (FC 2003)*, 2003.

[SW99] R. Safavi-Naini and H. Wang, “Multireceiver Authentication Codes: Models, Bounds, Constructions, and Extensions,” *Information and Computation*, vol. 151, pp. 148–72, 1999.

[SW98] R. Safavi-Naini and H. Wang, “New results on Multi-receiver Authentication Codes,” *Advances in Cryptology - EUROCRYPT ‘98*, LNCS 1403, pp. 527-541, 1998.

[WGL98] C. K. Wong, M. Gouda and S. Lam, “Secure Group Communications Using Key Graphs”, *Sigcomm ’98,* 1998.

[WHA00] D. M. Wallner, E. J. Harder and R. C. Agee, “Key Management for Multicast: Issues and Architectures”, RFC Editor, 1999.

[WL98] K. Wong and S. S. Lam, “Digital Signatures for Flows and Multicasts,” *IEEE ICNP 98*, 1998.

[Wang09] Qiyan Wang, H. Khurana, Ying Huang, and K. Nahrstedt, “Time Valid One-Time Signature for Time-Critical Multicast Data Authentication”, *IEEE INFOCOM*, Rio de Janeiro, 2009.

[ZF06] Y. Zhou, and Y. Fang, “BABRA: Batch-Based Broadcast Authentication in Wireless Sensor Networks,” *Proc. IEEE GLOBECOM*, 2006.

[ZF07] Y. Zhou, and Y. Fang, “Multimedia Broadcast Authentication Based on Batch Signature,” *IEEE Comm. Magazine*, vol. 45, no. 8, pp. 72-77, 2007.

[ZSJ03] S. Zhu, S. Setia and S. Jajodia, “Leap: efficient security mechanisms for large-scale distributed sensor networks”, in *ACM CCS, Washington D.C., USA*, 2003

[ZSW05] Z. Zhang, Q. Sun and W.C. Wong, “A proposal of butterfly-graph based stream authentication over lossy networks,” *in Proc. IEEE International Conference on Multimedia & Expo*, 2005

[ZSW+07] Z. Zhishou, J. Apostolopoulos, Q. Sun, S. Wee, and W. Wong, “Stream authentication based on generalized butterfly graph,” *In Proceedings of the IEEE International Conference on Image Processing (ICIP’07)*, 2007.

[ZZF10] Yun Zhou, Xiaoyan Zhu and Yuguang Fang, "MABS: Multicast Authentication Based on Batch Signature," *Mobile Computing, IEEE Transactions on*, 2010.