The Adaptive Multicast Data Origin Authentication Protocol

Liehuang Zhu*, Hao Yang, and Zhenghe Yang

Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application, School of Computer Science and Technology, Beijing Institute of Technology, China, Beijing
{liehuangz,haoy0320,zhenghey}@bit.edu.cn

Abstract. Most multicast data origin authentication schemes work under the fixed parameters without taking the problem of changeable network environment into account. However, the network conditions will obviously influence the efficiency of a protocol such as the time delay and the overhead. So adjusting the parameters adaptively to achieve the ideal state with the dynamic network is necessary. To achieve a high authentication rate and adapt to the changeable and unstable network environment, we proposed a multicast data origin authentication protocol which is adaptive depending on the packet error rates as well as the time delay, and is robust against the packet loss and injection. Our model can estimate a more appropriate packet error rate and make the time delay lower according to the feedback values got from the receivers using the Markov chain so that it can be adaptive. This strategy is especially efficient in terms of not only the adaptation of dynamic network but also the shortcut of overhead and delay.

Keywords: Authentication, adaptive, estimate, packet error rate, time delay, Markov chain.

1 Introduction

With the rapid development of broadcasting technology, it's applied to a lot of network applications for group communication. The constantly abundance and development of such applications makes the security and non-repudiation of group communication attract more and more attention. At the same time, the efficiency is also very important especially considering the low storage capacity nodes in the channel such as the mobile phone users as well as the dynamic network conditions.

In the group communication channel, it's probable that the data packets on the way be attacked by malicious participants or the adversary. So it's very necessary to guarantee the privacy as well as the non-repudiation. The digital signature can make sure

* This paper is supported by National Natural Science Foundation of China No.61272512 and Beijing Municipal Natural Science Foundation No.4121001.

J. Su et al. (Eds.): ICoC 2013, CCIS 401, pp. 109–121, 2013.

the non-repudiation of the scheme. However, the traditional way to sign every packet would make the computation and communication overhead very high. So the erasure code is included in to increase the authentication rate and decrease the overhead. Currently, most of the secure data origin authentication model is simply built on the hybrid model of the digital signature and the erasure code such as the SAIDA [1]. In the SAIDA scheme, the lost packets can be recovered in the range of m / n ($m < n$). What's more, the time delay is proportional to the parameter n and the overhead also increases with $1 / m$. It is obvious that do not adjust the parameters with the change of network will decrease the efficiency and may increase the pressure of low storage capacity nodes. So the service provider could adjust itself to the network environment dynamically to achieve the best effect becomes necessary.

However, most of the authentication schemes have no idea about adapting themselves to the variable network environment. The parameters in the schemes are always set at the beginning and never changed.

In this paper, we proposed an effective scheme to solve this problem. The Markov model is included in to make the system adaptive. The Markov chain can estimate the next state based on the states came before according to the state transition matrix, but it needs a lot of prior experience data to determine the matrix. According to a mass of time delay values and packet error rates before from the receiving nodes, the estimated ones can be given. So we combined the IDA, the Merkle HASH tree, as well as the Markov algorithm to achieve the adaptive and secure data origin authentication model. It can achieve the followed abilities:

- Resist the packet loss and injection. The IDA and Merkle HASH tree are combined to achieve the perfect non-reputation and can resist all kinds of packet attack from the network.
- Be adaptive to reduce the time delay. According to the feedback values, the parameter n will be adjusted to adapt the network environment and make the delay lower.
- Be adaptive to balance the overhead. The key parameter m / n can be estimated using Markov model to adapt the following network condition so that the communication and computation overhead is balanced.

2 Related Works

The TESLA scheme [2] proposed by Perrig realized the group data origin authentication by postponing sending the key of the MAC. This scheme has advantages of fast computation speed, less overhead and so on. However, TESLA needs the synchronous clock between the sender and the receivers, and it's difficult to be guaranteed under the open network environment.

Park proposed the SAIDA protocol [1] which can disperse the hash values and the digital signatures of all packets of one block according to the Information Disperse Algorithm (IDA). The receivers could recover the hash value and the digital signature

messages as soon as received parts of the packets, and in this way the loss of signature packets is solved. Lysyanskaya proposed the LLT protocol [3] which firstly solved the forged packets injection problem using the Reed-Solomon error correcting code, but it needs more computation overhead.

An authentication scheme based on the Reed-Solomon erasure code and the one-way hash function was proposed by Anna Lysyanskaya et al. in 2010 [4]. Similarly, it calculates the signature of the whole group and disperses it using the RS erasure code. This protocol adds only one signature on one group and decreases the computation and communication cost, but it needs $O(n^2)$ time delay between the participants.

In 2010, an optimized scheme based on Merkle tree as well as TESLA was presented by Yang Li [5]. It uses the Merkle tree to authenticate and use delayed disclosure of keys in TESLA algorithm to ensure authenticity of message. This scheme not only obviously decreased the storage cost, but also could be compatible with complex network environment and treat burst loss well. But it needs high computation overhead especially for the receivers.

Then Seyed Ali Ahmadzadeh gave a scheme based on the geometrical model named GMAC [6]. It maps the hash values of data packets in one group to a vector space with n degrees to filter the illegal packets. The cost of this protocol is far less than PRABS and can resist the packet loss and injection, but it has a high computation complexity due to the use of geometrical model.

In 2012, Yongsheng Liu et al. [7] proposed a kind of signature dispersal authentication scheme based on the PKC. It calculates only one ECC digital signature for one group and then disperses it into all the packets in one group. It costs little communication and computation cost whereas the time delay cannot be avoided.

Kannan Balasubramanian et al. proposed the HTSS scheme [8] in 2012. It generates the keys by the hash tree and after signing the messages, it divides the signature to the packets in its period with the SDA. This protocol decreases the overhead added to every packet and can resist the packet loss well. However, it has nothing to do with the packet injection or forgery.

Hong Tang et al. proposed a kind of broadcasting data origin authentication protocol called EPJRSA [9] based on the Merkle HASH tree in 2008. This protocol combines the erasure coed as well as the Merkle HASH tree, and by adding all the brother nodes' hash values in the tree to one packet, it can resist the packet injection as well as the packet loss. This guarantees the reliability of the authentication. But it also increases the communication cost and the time delay.

Gaolei Fei and Guangmin Hu proposed the unicast network loss tomography based on k-th order Markov chain in 2011 [10]. This protocol introduces the k-th order Markov chain (k-MC) to describe the link packet loss process, and then uses the pseudo maximum likelihood protocol to estimate the state transition probabilities of the k-th order Markov chain. When the k is large enough, this protocol can be capable of obtaining an accurate loss probability estimate of each packet based on unicast end-to-end measurements. However, this protocol can only be used in the unicast network and the computation overhead is high.

3 Our Adaptive System Model

3.1 The Robust Data Origin Authentication Model

Our protocol combined the IDA and the Merkle HASH tree to construct a kind of trusted data origin authentication model [11] that can resist both the packet loss and the pack injection within the threshold values.

The IDA [12] is constructed with two important parameters n and m. The file to be encoded can be segmented into n pieces. And only m pieces of File are given, we can reconstruct File according to the steps of IDA.

In our method, firstly two important encode parameters have to be set: n and m. Then we use the Merkle HASH tree as the base framework, dividing a block of data into n packets and then construct the Merkle HASH tree as Fig. 1.

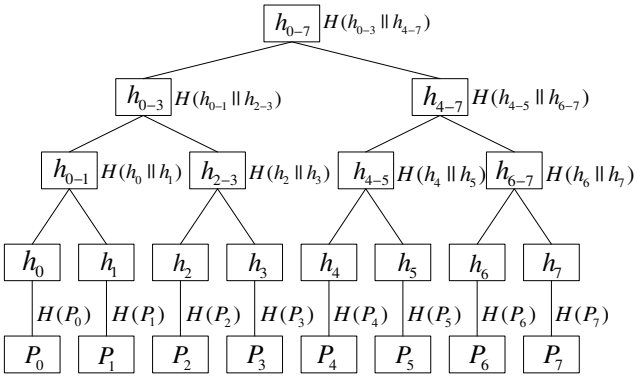

Fig. 1. The Merkle HASH tree of 8 nodes

Secondly we get the concatenation value $F = H(P_0) \| H(P_1) \| \ldots \| H(P_{n-1})$ and the signature of the group $\sigma(K_{public}, H(F))$. Next using IDA, we can encode and disperse both the concatenation value and the signature into n pieces as $\{F_0, F_1, \ldots, F_{n-1}\}$ and $\{\sigma_0, \sigma_1, \ldots, \sigma_{n-1}\}$. So a complete packet i includes three parts as Fig. 2, the data block P_i itself, the hash and signature segments F_i and σ_i, and all the related hash values of the brother nodes from it to the root in the Merkle HASH tree, such as $\{h_0, h_{2-3}, h_{4-7}\}$ to packet P_1 in Fig. 1.

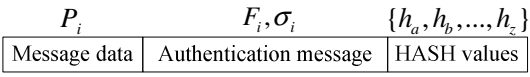

Fig. 2. Three parts of one packet

According to the IDA, as long as m in the n pieces of F_i or σ_i received, the intact hash value or signature can be recovered, and then the data can be authenticated. So it can tolerate the packet error in the range of m. Then for the packet injection or forgery, the Merkle HASH tree can guarantee that the forged packets are recognized based on

the third part of one packet. Every packet received can recompute the hash value of the root using the third part, so the different group with fewer ones will be discarded.

3.2 The Estimate Model Using the Markov Chain

We apply the Markov chain to estimate the incoming packet error rate based on the previous ones. As we all know, the Markov chain describes a kind of state sequence that every state in the sequence depends on the previous finite ones according to the state transition matrix constituted by the transition probabilities. So what we want to get is the transition matrix of the data packet error rate. At the same time, we adjust the time delay based on the feedback values.

The Adjustment of Block Size Parameter

We can know from the IDA that the more packets included in one block, the smaller overhead each packet will take. On the other hand, the time delay will increase with n. So the balance between these factors is necessary.

Here we define a threshold value of the time delay as t, as well as a threshold limit time t_0. Only if the average time delay in one period exceeds the threshold value t, the size n will be decreased. And if the time delay keeps under the threshold value for t_0 time, then we will increase the size n.

Then we also need to consider the overhead with the value n. It should be controlled so that the overhead per packet would not be too big. In our experiments, the n should be no more than 128 considering the limit of the experiment environment.

The Estimate of Packet Error Rate

First of all, we define k states which present the default values of packet error rate as O_1, O_2,…, O_k ($0 \le O_i < 1$ and $1 \le i \le k$). And the k values are set as the coordinates of the transition matrix, in which the a_{ij} ($1 \le i, j \le k$) presents the probability count of transition between the rates as (1). For example, a_{2k} means the situation that the previous packet error rate is O_2 and the next one is O_k appears a_{2k} times in this period.

$$\begin{array}{c} \\ O_1 \\ O_2 \\ \\ O_k \end{array} \begin{array}{c} \begin{array}{cccc} O_1 & O_2 & & O_k \end{array} \\ \begin{bmatrix} a_{11}, a_{12}, ..., a_{1k} \\ a_{21}, a_{22}, ..., a_{2k} \\ \vdots \\ a_{k1}, a_{k2}, ..., a_{kk} \end{bmatrix} \end{array} \tag{1}$$

What should be noticed is that k must be an integer in the range of $(0, n]$. The bigger the k is, the more accurate the matrix is but the estimate may be inaccurate with too many $a_{ij} = 0$ in the matrix. The smaller the k is, the more accurate the estimate is. However, the estimate may be limit within several values. So the k is better fixed in the range of [8, 32] to get a good result in the following experiments. And the values

of O_i ($1 \le i \le k$) should be chosen evenly in [0, 1), so that all rates can be assorted to one nearest state O_i. Then we get a state transition matrix whose scale is $k \times k$.

An example is given. Suppose we have $n = 128$ packets in one block, then the data packet sequence can be donated as $P_0P_1\ldots P_{127}$. And at the clients, the correctly received ones are signed as 1 and the others are signed as 0. Then the packet error rates Q_js ($Q_j = N_0 / n$, N_0 presents the number of $P_{0\sim n-1} = 0$ in the j-th sequence) can be given. Here we set the scale $k = 8$ and the O_is of the transition matrix are 0, 1/8,…, 7/8 (chosen evenly in [0, 1)). If the adjacent two packet error rates Q and Q' got from the feedback strings are 75/128 and 23/128, they will be respectively assorted to the nearest states $O_6 = 5/8$ and $O_2 = 1/8$. Then the count a_{62} in the matrix should plus 1. So we fill the transition matrix as (2) after all the rates are assorted to the nearest O_i in one experiment. After that, it will search the matrix according to the following rate Q_A, assumed to be assorted to state O_7, to find out the maximum probability $a_{7Max} = a_{74}$ to reach the next state $O_4 = Q_B$. Then Q_B is the estimate the Markov chain model made.

	0	1/8	2/8	3/8	4/8	5/8	6/8	7/8
0	7	1	1	8	53	38	9	1
1/8	8	1	0	1	1	35	34	2
2/8	19	2	1	1	0	1	1	27
3/8	55	5	2	2	0	1	1	1
4/8	24	41	6	2	1	0	0	1
5/8	4	31	41	4	2	0	0	0
6/8	0	0	23	32	8	3	4	0
7/8	1	0	0	17	10	4	0	0

(2)

So in this way, we can estimate the incoming possible packet error rate according to the feedback values. All the probability values a_{ij} ($1 \le i, j \le k$) in the state transition matrix should be got from a mass of feedback values in one period and are believable.

3.3 The Adaptive Multicast Data Origin Authentication Model

In our model, we set three kinds of end as the server, the adversary and the clients. The network model is set as Fig. 3. The server broadcasts messages to the heterogeneous clients and then the chosen clients will respond the bit strings which represent whether the packets received correctly or not as well as the time delay messages. The adversary could control parts of the network and attack the data packets on the way.

At the server, firstly some initial parameters are determined: the encoding parameters n ($n = 2^p$, p is a positive integer) and m ($m < n$), a set of k values of packet error rate O_i ($0 \le O_i < 1$, $1 \le i \le n$) as the coordinates of the transition matrix, and then three threshold values, the time delay threshold value t, the limit time t_0 and the packet error rate threshold value β ($0 < \beta < 1$). The same parameters are shared at the clients too.

Secondly, the server starts to broadcast the message packets and the authentication messages encoded by the IDA and the Merkle HASH tree as Fig. 2 with the initial parameters n and m. The clients would send feedback messages of the time delay and packet error rate to the server.

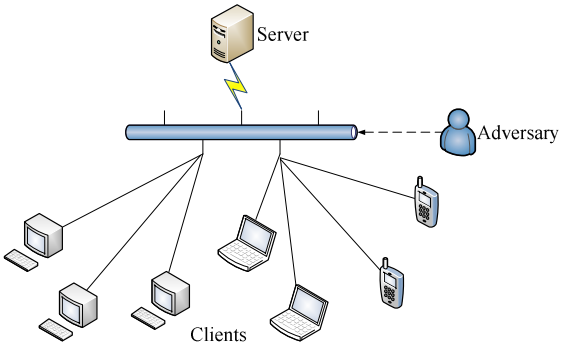

Fig. 3. The multicast framework model

If the average decoding time in one period is beyond the threshold value t, the representative clients will send the server a timeout label, and then the server will decrease the value of n as $n = n / 2$. On the other hand, if the average time delay keeps under the threshold value t for t_0 time, then the server recovers the size n as $n = n \times 2$.

And if the packet error rate is beyond the threshold value β, for example, the lost packets and the forged packets from the adversary are altogether ω in a block of n packets, and $\omega / n > \beta$, then the representative clients will send the server the packet sequence bit strings $P_1P_2\ldots P_n$. And the state transition matrix can be filled up using our protocol as (1).

Next the server will estimate the incoming packet error rate according to the state transition matrix. It will get a maximum probability to reach the next state O with the last received rate, and the rate $Q = O$ is the result estimated by the Markov chain. At the same time, after the messages of one block received, the clients do the decoding operation to the packets and authenticate the messages.

At last, a new encoding rate m / n is determined by the server according to the estimated packet error rate Q ($m / n = 1 - Q$). With the block size parameter n adjusted by the timeout threshold value, a pair of new values of n and m will be used.

4 Experiments and Results

We do our experiments under the local area network environment, and control the network as three kinds of packet error model: random, stable and burst. One end broadcasts the messages as the server, one end simulates the adversary to attack the channel randomly, and other ends receive as the clients.

In our experiments, the server firstly sets $n = 128$, $m / n = 1/2$, and the transition matrix scale as $k = 8$ ($O_i = 0, 1/8,\ldots, 7/8$). Then it broadcasts a message, and the adversary randomly attacks. The clients will receive the attacked packets and authenticate them. During this period, the chosen nodes will send the feedback to the server (here we ignore the error threshold and send every packet error rate back). The server needs to gather the feedback values and then do the Markov estimation every 10 minutes. Also the time delay values are calculated and sent back in the same way.

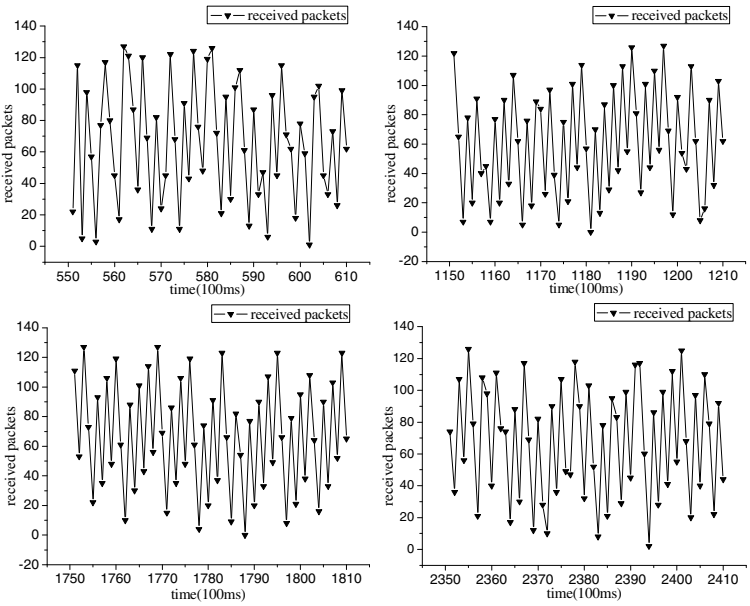

Fig. 4. The error rates under random condition

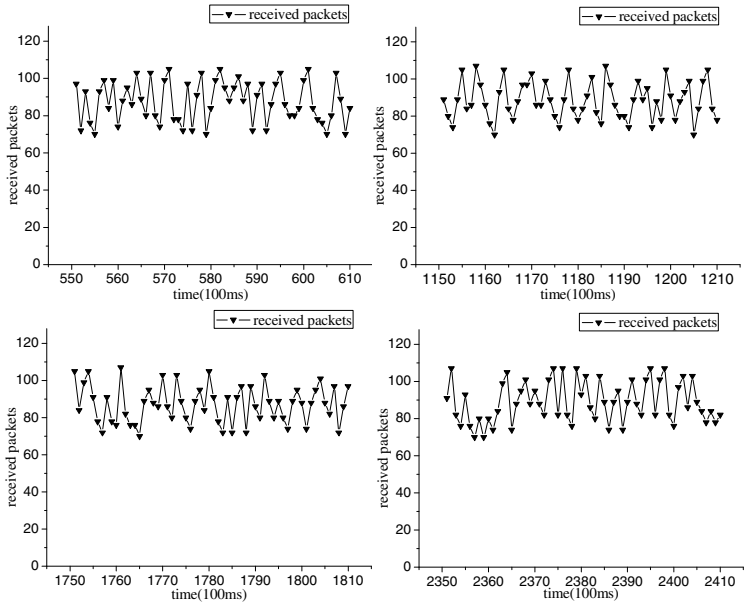

Fig. 5. The error rates under stable condition

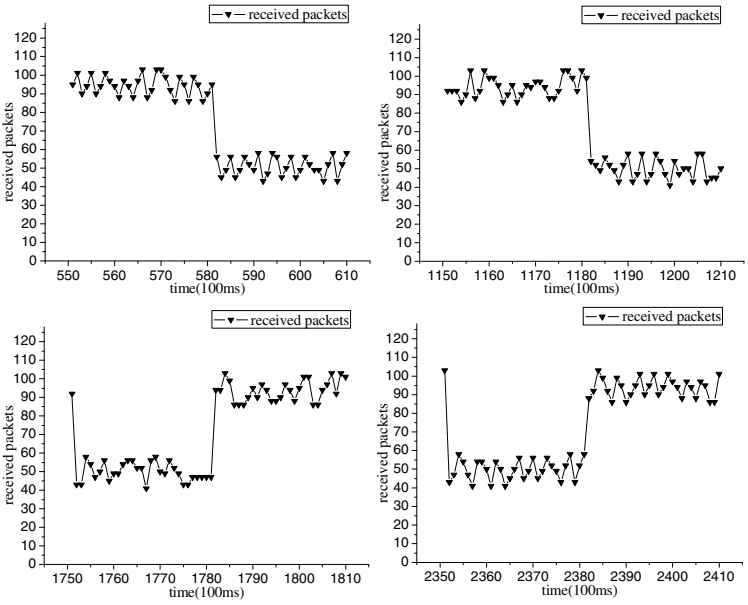

Fig. 6. The error rates under burst condition

In the random error model, we can see from Fig. 4 that the error rates vary with time randomly. So we estimate every 10 minutes. And the following $m / n = 1 - Q_e$ values can be given.

Also we can see from Fig. 5 and Fig. 6 that our scheme works well under the stable as well as the burst error conditions. The network is relatively stable in the stable model which means that the error rate keeps within a permissible range. In the burst error model, the error rate may vary suddenly at any time for the changeable network.

Under the stable network environment, the m / n parameter is also relatively stable as Table 1. On the other hand, if the network condition changes suddenly, the rate may be varied obviously and frequently as Table 2.

And our scheme can work very well with different n values. We can get this conclusion from Table 3 (the initial m / n rates are all 1/2).

Table 1. The estimated error rates under stable network environment

the values of *n*	**the estimated new *m* / *n* values**								
128	4/8	4/8	3/8	3/8	4/8	3/8	4/8	4/8	3/8

Table 2. The estimated error rates under unstable network environment

No.	*n* value	the estimated new *m* / *n* values								
1	128	8/8	7/8	7/8	1/8	2/8	6/8	8/8	5/8	8/8
2	128	7/8	2/8	2/8	8/8	8/8	4/8	8/8	7/8	5/8
3	64	3/8	8/8	8/8	3/8	5/8	6/8	8/8	8/8	8/8
4	64	8/8	7/8	8/8	6/8	7/8	3/8	6/8	7/8	3/8
5	32	5/8	5/8	8/8	5/8	3/8	8/8	2/8	5/8	5/8
6	32	2/8	8/8	7/8	8/8	8/8	2/8	5/8	7/8	8/8

Table 3. Several estimated results

No.	*n* value	the estimated new *m* / *n* values								
1	128	7/8	2/8	2/8	8/8	8/8	4/8	8/8	7/8	5/8
2	128	4/8	6/8	2/8	7/8	8/8	2/8	6/8	4/8	3/8
3	64	3/8	8/8	8/8	3/8	5/8	6/8	8/8	8/8	8/8
4	64	8/8	7/8	8/8	6/8	7/8	3/8	6/8	7/8	3/8
5	32	5/8	5/8	8/8	5/8	3/8	8/8	2/8	5/8	5/8
6	32	2/8	8/8	7/8	8/8	8/8	2/8	5/8	7/8	8/8

The comparison of overhead between the fixed scheme and the adaptive scheme under the same network environment is given in Fig.7.

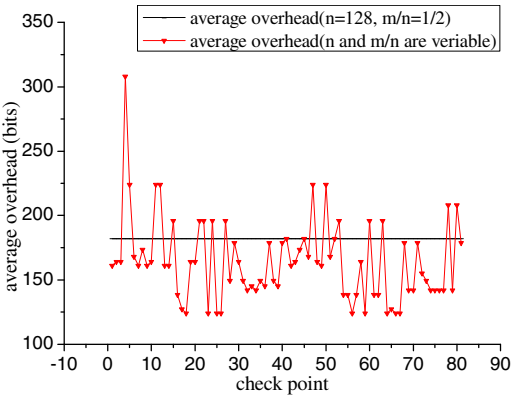

Fig. 7. The overhead comparison

We can see from it that most overhead of the check points in the adaptive scheme in which the parameters vary with the network environment is much lower than the fixed parameters scheme (n = 128 and m / n = 1/2). And the average overhead of the whole check points is 164.205, which is less than 182 of the original scheme.

Also, the superiority of the verification rate of the adaptive scheme is obvious too as Fig. 8. With the changeable packet error rate which varies randomly between 0 and 1 and the fixed initial parameters m / n, the verification rate of the system is not as expected. On the other hand, the adaptive scheme can achieve an ideal verification rate in any condition.

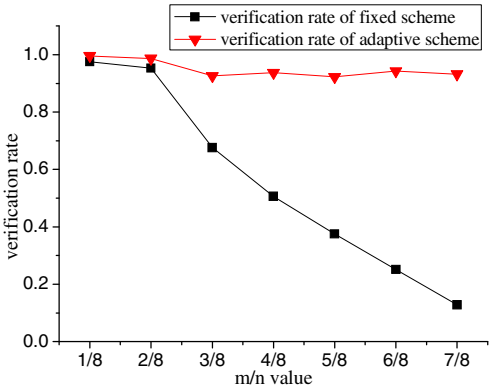

Fig. 8. Verification rate under unstable environment

At the same time, the chosen nodes will count the decoding time of every block and compared to the threshold value t. Here we set t = 250ms and t_0 = 103ms, so we can get the Table 4 as followed (the bold italic column presents the time t_0).

Table 4. The time delay values changing table

No.	*n* value	the time delay near the critical point (ms)					
1	128	276.4	313.7	263.2	***290.1***	203.3	236.1
2	128	234.7	229.2	227.8	***231.6***	303.4	343.7
3	64	225.9	198.3	246.1	***209.3***	258.0	269.4
4	64	257.3	254.9	273.4	***289.0***	226.1	231.8
5	32	193.2	197.7	215.2	***220.4***	267.9	258.5
6	32	265.9	282.3	280.7	***274.1***	238.4	240.2

We can get from the Table 4 that when the time delay values exceed the threshold value t for t_0 time, then the value of n would be decreased to reduce the time delay (the 1st, 4th and 6th rows). Otherwise the delay values keep under the threshold value t for t_0 time, the value of n would be increased (the 2nd, 3rd and 5th rows) as Fig. 9.

5 Conclusion

We can see from the experiment results that our protocol is efficient in adapting itself in all kinds of network environment especially when the network changes frequently.

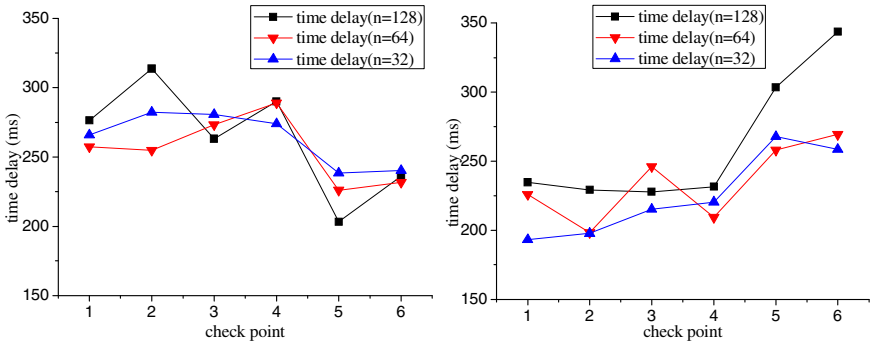

Fig. 9. Time delay decreases/increases with $n = n / 2$ at the check point 4

Our scheme achieved the highest verification probability with less overhead within a certain range and adjusted the parameters dynamically in the changeable network environment which the other schemes cannot solve.

And obviously, our scheme might not be appropriate in situations where the data to be sent is generated in real time, and immediate broadcast of it is crucial. Our scheme will be most useful in situations where it needs high authentication rate and efficiency with less overhead but the network varies frequently and irregularly.

References

1. Park, J.M., Chong, E.K.P., Siegel, H.J.: Efficient multicast packet authentication using signature amortization. In: Proc. IEEE Symposium on Security and Privacy, pp. 227–240 (2002)
2. Perrig, A., Canetti, R., Tygar, J.D., Song, D.X.: Efficient authentication and signing of multicast streams over lossy channels. In: Proc. IEEE Symposium on Security and Privacy, pp. 56–73 (2000)
3. Lysyanskaya, A., Tamassia, R., Triandopoulos, N.: Multicast authentication in fully adversarial networks. In: Proc. IEEE Symposium on Security and Privacy, p. 241 (2004)
4. Lysyanskaya, A., Tamassia, R., Triandopoulos, N.: Authenticated error-correcting codes with applications to multicast authentication. In: Proc. ACM Trans. Inf. Syst. Secur. (2010)
5. Li, Y., Zhang, M., Guo, Y., Xu, G.: Optimized source authentication scheme for multicast based on Merkle Tree and TESLA. In: Proc. Information Theory and Information Security (ICITIS), pp. 195–198 (2010)
6. Ahmadzadeh, S.A., Agnew, G.B.: Poster: a geometric approach for multicast authentication in adversarial channels. In: Proc. ACM Conference on Computer and Communications Security, pp. 729–732 (2011)
7. Liu, Y., Li, J., Guizani, M.: PKC Based Broadcast Authentication using Signature Amortization for WSNs. In: Proc. IEEE Transactions on Wireless Communications, pp. 2106–2115 (2012)
8. Balasubramanian, K., Roopa, R.: HTSS: Hash Tree Signature Scheme for Multicast Authentication. In: Proc. International Conference on Recent Trends in Computational Methods, Communication and Controls (ICON3C 2012), pp. 28–32 (2012)

9. Tang, H., Zhu, L.: Efficient packet-injection resistant data source authentication protocol for group communication. Proc. Journal on Communications 11A, 96–100 (2008)
10. Fei, G., Hu, G.: Unicast network loss tomography based on k-th order Markov chain. Proc. Journal of Electronics & Information Technology 33(9), 2278–2282 (2011)
11. Yang, H., Zhu, L.: The application of data origin authentication to video multicast communication. In: Proc. International Conference on Multimedia Technology (ICMT), July 26-28, pp. 5129–5132 (2011)
12. Rabin, M.O.: Efficient dispersal of information for security, load balancing, and fault tolerance. Proc. J. ACM, 335–348 (1989)