A Light-weight Authentication Scheme in the Internet of Things using the Enhanced Bloom Filter

Xiaoyan Huo
Information Construction and Management Center, Jiaozuo University, Jiaozuo, Henan, 454003, China

***Abstract*—Authenticated key exchange mechanisms are critical for security-sensitive Internet of Things (IoT) and Wireless Sensor Networks (WSNs). In this area, the Bloom Filter (BF) plays a crucial role directly and indirectly, which has a significant advantage in space and time. Light-weight input authentication is one of the most challenging tasks in IoT. Weak or inefficient defense algorithms can allow fake information to enter the system, share information, send unnecessary messages, and reduce network efficiency. The utilization of an augmented Bloom filter for creating an authentication prominent called En-route Authentication Bitmap (EAB) has a substantial advantage over traditional methods that involve direct usage of Message Authentication Codes (MAC). This effective method of EAB picks the fake information almost accurately, thereby reducing the feeding attacks within not more than two steps taken by the attacker. EAB necessarily needs only a few bytes of bandwidth for efficient defense against at least ten forward steps of the adversary. Without hesitation, the Augmented Bloom filter and its components are becoming more common in network defense mechanisms.**

Keywords—En-route authentication bitmap; message authentication codes; internet of things; bloom filter

I. Introduction

As the Internet of Things (IoT) expands, it is poised to transform human lifestyles and release several monetary benefits [1, 2]. Security and trust issues present significant adoption limitations for the IoT [3]. Blockchain, a distributed and tamper-resistant ledger, maintains regular statistics data at only places and can deal with the information protection situation in IoT networks [4]. The IoT is an entirely novel paradigm combining technologies and elements that come from different strategies [5]. The real and virtual worlds constantly interact through embedded devices, communication technology, sensing technology, Internet protocol, pervasive computing, and ubiquitous computing [6]. The IoT vision relies on smart objects. Putting intelligence into everyday objects enables them to collect environmental data, interact with the physical world, and interconnect via the Internet with alternate facts and records [7]. New business opportunities will arise as devices become increasingly connected and records become increasingly plentiful. This enables delivering tangible advantages to individual residents, the economy, the environment, and society [8].

Designers are primarily concerned with the trade-off between cost, computation, and security [9]. Although end-to-end authentication can be validated by sharing keys between the source and target, accuracy is complicated halfway, where intermediate nodes cannot determine authenticity without additional information [10]. In wireless broadcasting, promotional messages should be eliminated as much as possible [11]. Data passes through a node route every time. The relative nodes are instructed to remain still and observe the information even though those nodes are neither the target nor the transporters. Making things worse, the transfer packet data does not make use only the way where the neighbor nodes are present but also the neighbor region [12].

There is competition between the two-hop nodes for communication channels, similar to the hidden ones problem [13]. Hop-by-hop routing easily identifies entryways for the attackers to feed and take the system information [14]. The endpoint cannot be made to believe the take information by the adversary unless the two-end key is weak or unsafe [15]. Providing false messages in the pathway, redundant traffic creation, waste usage of computation power of the route, and retarding the efficiency of the network can usually be made simple by the attacker [16]. Using a network-wide authentication key is a natural way to solve this problem. However, the adjustment of one node can lead to the adjustment of the whole arrangement. It can be seen as the most effective way to follow µTesla, in which the authentication key is updated frequently. Nevertheless, this method has its advantage in that network synchronization with standard timing error should be present. Synchronizing the timing is complicated if the route is extensive. Moreover, these approaches are suitable for central nodes, like base stations [17].

Attaching all Message Authentication Codes (MACs) with the source node to the payload for every router is another effective solution [18]. As the number of route nodes increases, the volume of the message gradually increases. These problems necessitate the need for lightweight en-route authentication schemes in multi-hop networks. This paper discusses a statistical way to solve the hop-by-hop message authentication problem between two ends. We efficiently examine data filtration in a routing way for bandwidth mitigation. En-route Authentication Route Bitmap (EAB) is a structure of the bloom filter method that is appended to appropriate messages as extra data. We propose a method for protecting the system from attack by injecting bogus data since the damage is contained in its immediate area. Slight numerical adjustments can improve efficiency and protection.

The challenges of IoT security are securing restrained devices, empowering devices, validating devices, updating devices, ensuring the confidentiality and reliability of data, managing web, mobile, and cloud functions, guaranteeing high availability, detecting vulnerabilities and incidents, handling vulnerabilities, and forecasting security issues. To overcome the constraints in the IoT, we propose a system that can defend the Internet of Things from attack and implement a lightweight authentication. Bloom filters are primarily responsible for authentication. This paper implements an Augmented Bloom Filter. The EAB is inaccurate for mapping multiple MACs. EAB classifies false data during the first constrained hop instead of being dropped at the first sign. Therefore, the false negative rate adjacent to the adversary is satisfactory, but it quickly rises with increasing bound counts. This swapping drastically reduces authentication communication overhead.

Multihop wireless networks face challenges with lightweight en-route authentication. False data can be injected into a system in the absence of an efficient defense mechanism, causing redundant message forwarding and consuming node power. We create an authentication manifest using EAB, which differs from conventional approaches that directly use MACs. By filtering out false data with a high success rate, EAB makes injection attacks less likely to spread beyond one or two hops. While EAB spends a few bytes of bandwidth, it can effectively protect tens of hops along the forwarding path.

The rest of the paper is organized in the following manner. Preliminaries are presented in Section II. Basis design of the mechanism is discussed in Section III. A discussion of random padding improvements is presented in Section IV. Section V concludes the paper.

II. Related work

Nianmin, Haifeng [19] proposed a method for ensuring data integrity. This one-level hash structure has three advantages over traditional hash trees: a lower computation overhead, a lower space overhead, and the ability to adjust the security level. The proposed method was compared with an efficient integrity checking scheme for its overhead and security. According to the evaluation results, the method has a lower overhead for most benchmarks.

Saravanan and Senthilkumar [20] present an enhanced bloom filter technique to represent large sets of data in a secure manner and apply it to distributed applications such as web caching and peer networks. Also, to limit the possibility of unauthorized access to the dataset by unauthorized intruders, the enhanced bloom filter is applied with an upper bound on the false-positive probability by increasing its capacity as the packet data size increases. As compared to the existing dynamic bloom filter approach, the experimental results show a 42.5% improvement in packet data security.

A scheme based on an improved counting bloom filter was developed by Wang, Wang [21] for securing authentication keys in heterogeneous sensor networks. By applying the Set theory to heterogeneous sensor networks, the counting bloom filter algorithm is improved to address the authentication key agreement problem. As demonstrated by experimental results, the proposed scheme has greater network scalability, lower communication costs, and can resist brute force attacks during a node capture.

Malhi and Batra [22] proposed an authentication framework based on pseudonyms that preserves privacy. An ID-based signature scheme is used for vehicle-to-RSU communication, and a new digital signature scheme is designed for vehicular communications. The identity of the vehicle is revealed by multiple authorities in the event of revocation. Bloom filters are used to improve the signature verification scheme, which was implemented on a simulated environment to evaluate the results.

Mbarek, Sahli [23] developed an efficient authentication method to authenticate sensors utilizing the Bloom Filter. DoS attacks can be mitigated by using multiple MACs along with a Bloom filter in delayed key disclosure schemes. A set of scenarios was used to assess the feasibility of the protocol. Authentication protocol reduces false positives in Bloom Filters, according to the results.

By derived from the cryptographic permutation Xoodoo, Sateesan, Vliegen [24] proposed a new noncryptographic hash function, called Xoodoo-NC, for developing ultra-high-speed Bloom filters on FPGAs. The Xoodoo-NC hash function inherits the desired avalanche properties of Xoodoo and the low logical depth, resulting in an ultra-low-latency non-cryptographic hash function.

III. Preliminaries

A. System Model

There are several computing nodes in IoT. These nodes can pick the links between themselves and their adjacent nodes and set up a connection with nodes that are far away by implementing a path through the nodes in the way. There may be several server nodes in existence containing effective resources that may include cloud computing and fog computing. Node A and Node B can be considered fellow nodes if the authentication key is shared. These authentications between fellow nodes that are end-to-end are certified with contributed keys through some systems. To share information between two end nodes, both nodes should possess a common key, or the sender should appoint the information to be delivered to both end nodes. The division of the path is considered a series of transmission systems. Using step-by-step verification as a benefit of end-to-end information sharing is mainly discussed here. Considering the routing paths to be constant, we assume it to be sensed by the node from the origin.

B. Threat Model

End-to-end authentication usually assures the security and purity of the information. This security can be breached by exposing the pairwise key among the source and target nodes. Imposing false traffic and message authentication on the nodes' path and diminishing the network's efficiency is the attacker's main aim in this case. Here the adversary creates upstream packets or forge packets and spreads them in the route path. These can falsely act as the origin or pathway of some other fake data. Rejection of messages that come across the path is an alternative attack technique of the adversary. This type of attack naturally does not have the right solution until there is an effective connection between the relative nodes.

C. Bloom Filters

Bloom Filters are probabilistic data structures with some errors. It is a simple hashing algorithm that requires very little memory. Various research areas have been applied by Bloom Filter to boost performance. As a membership filter, it returns either "true" or "false". The term "true" can, however, refer to either a false positive or true positive. In the same way, a "false" can also be a true negative or a false negative. False positives and false negatives are errors of the Bloom Filter. However, the error is negligible and tolerable. Nevertheless, Bloom Filter is not suitable for many systems, such as real-time systems.

A very less false positive and null false negative is effectively obtained through the Bloom filter, which is greatly useful in representing a pack of inputs to fulfill the above criteria. Firstly, every bit used in the Bloom filter remains unmarked, and k is set to be stochastic hash functions. H $=\{H^1,\ldots\ldots k\}$ where the range is from $\{0,\ldots\ldots,m-1\}$. Representation of the set $S=\{s_1\ldots..s_n\}$ of a number of inputs (n), n and H_i (s_j) are set to 1 where l<=j<=n and l<=I<=k. Where every input s_i are set to 1 to the bitmap, the chances for a particular bit to be marked is $p=1-(1-1/m)k_n$. It results in nearly $mp=(1-(1-1/m)^{kn})$ bits in the map being marked averagely. The discussed method is effectively used to create a Bloom filter, where $Bitmap \xleftarrow{bloom} H, S$.

In order to evaluate a new input x in S, a mark for every hi(x) bit is considered. If everything is marked, then it can be definite that x is input in S. Here, the only flow along with the small false positive chance f=(1-p)^k is probable, based on the value of m/n and k. If anything is left unmarked, x is definitely not present in S.

Border and Mitzenmancher coined the Bloom filter. Using a list or sets at premium spaces, using the Bloom filter is relevant, where it is possible to lower the severity of false positives. Augmented Bloom filters are more efficient when compared to other set representation data structures such as hash tables, binary search trees, linked lists, or simple arrays. Here we use a bloom filter array for a defense mechanism. The augmented bloom filter proceeded with two main steps, insertion operation and membership testing. Firstly, the elements in the set S should be inserted into the bloom filter array of m bits. Let the elements be x1, x2, x3,…, xn, where n is the number of elements. This makes it easier for members to test whether the element is present in the set or not.

D. *Insertion Operation*

The elements in the set S are added to the bloom filter array after yielding corresponding hash keys of individual elements using hash functions such as MD5. Here we are using a hash function named MD5 algorithm. The usage of the hash function greatly reduces the occurrence of false positives. In contrast, using a single hash function to insert elements in the bloom filter array leads to a high possibility of false positives. There will be no false negative outputs either by multiple hash functions or by implementing the hash function differently into the bloom filter.

Usually, the bloom filter implements three hash function algorithms to obtain three different key values. Here we will utilize a single hash function, but a slightly different method is preferred. Initially, divide the input value into three parts and then hashed by using the MD5 algorithm individually, which can be represented as $hk_1(x_1)$, $hk_2(x_2)$, $hk_3(x_3)$, where k_1, k_2, and k_3 are the hash functions. In order to generate hexadecimal values using the MD5 algorithm, the following steps must be sequentially applied.

1) Appending padding bits: In the first step, the element x, initially a URL, is converted to bits. By adding a single '1' bit at the end of the 'b' bit message, the message becomes divisible by 448 or 512.

2) Appending length: By adding a 64-bit representation, an output multiple of 448 can be converted into a multiple of 512.

3) Buffer initialization: This step divides the b-bit result from the preceding step into four 32-bit registers (A, B, C, D). A 128-bit message digest is derived using these registers.

4) Processing the message: Four auxiliary functions are applied to the message, and various processing steps produce the required output. The plain text is then converted into cipher text, resulting in the message digest.

We convert the obtained hexadecimal values into binary values, which can finally be inserted into the bloom filter, where the bits are initially zero. After adding the elements in the set, those ‘0’ bits on the corresponding positions will eventually become ‘1’ based on the inserted element.

E. *Membership Testing*

We check their similarity, ensuring it is already present in the array. Hence the bits need not be stored in the Augmented Bloom filter array. If not, we should add those bits to the Augmented Bloom filter array after checking whether the current bit is 1. If the relevant Augmented Bloom filter array is 0, it changes to 1. Otherwise, nothing is changed in membership testing.

1) Augmented bloom filter array: The Bloom filter is a space-efficient probabilistic statistics structure for determining whether elements belong to sets or not. They are astonishingly honest: take an array of k, n, m, and p. Both are examined for the use of hash features. By setting all bits, the element possibly now exists via a false positive rate of p; if the number of the bits is not specified, the element genuinely does not exist. Bloom filters locate an in-depth type of usage, counting and tracking articles that one has to examine, accelerating Bitcoin clients, detecting malicious links, and improving cache performance. This will help us select the most reliable size for our filter. Fig. 1to 3 show graphs among p versus n, m, and k. n refers to the number of items within the filter, p stands for the probability of false positives, the fraction between zero and one or various representing 1-in-p, m signifies the number of bits in the filter out, and k shows the number of hash functions. These values are computed by Eq. 1-4.

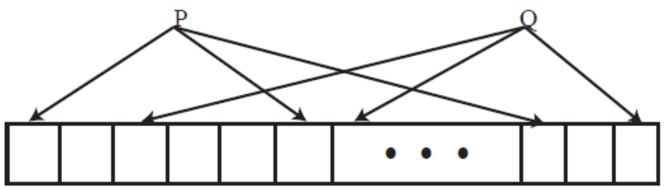

Fig. 1. Dynamic bloom filter.

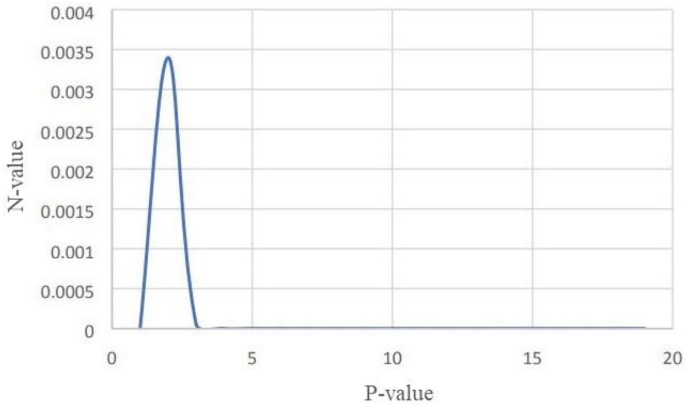

Fig. 2. Graph p vs. n.

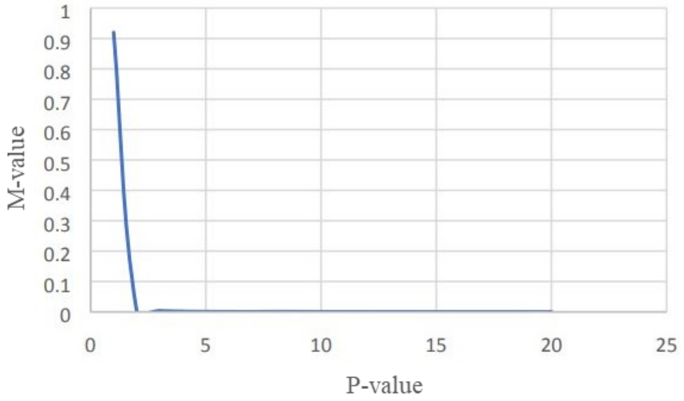

Fig. 3. Graph p vs. m.

Consider the Bloom Filter in which P and Q are two input elements. In Bloom Filter, k represents the number of hash functions of a key or element. In Figure 2, P and Q occupy 3 cells if k = 3. P is hashed into three different places in the array. In some Bloom filters, fingerprints are stored, while in others, only '1' or '0' are stored. Also, Q is hashed three times in the array. A positive answer of the Bloom Filter requires those three positions to be set (true) when checking the membership of P or Q.

F. *Overall Work*

In situations where non-symmetrical methods such as digital signatures are barely suitable, message authentication codes are commonly implemented in end-to-end authentication. They are most commonly provided in HMAC format. This step-by-step authentication is greatly relevant to broadcast authentication as a privacy enhancer. Hence, WSNs in cellular networks are mostly supplied with this system. Here this system mainly aims to send information to a particular target. Broadcast authentication is very good for securing hop transactions, as every cellular network depends on the broadcast. Delivers a method of hop-by-hop authentication that greatly assures the detection of any imparted false messages by the sink node, where ignoring nodes never crosses the limit. The target should contact adjustment nodes to connect their message authentication codes. Borgia [25] provides a key chain method for LHAP [26], where another keychain is generated to evaluate information from adjacent nodes. Here, consistency assurance among neighbors is essential. In the case of HEAP [27], every node gets message authentication codes connected, which would increase the band in turn. It is compulsory when ALPHA [28] is concerned with good relations between the source and the target. Even though Curtain [29] becomes the first want to use the bloom filter, authentication of broadcast from the origin is the authors' concern. We have successfully connected the message authentication codes of neighbor nodes with the help of Bloom filters, which can be as efficient as theirs.

$$k = \text{round}\ ((m / n) * \log (2)) \tag{1}$$

$$m = \text{ceil}\ ((n * \log(p)) / \log (1 / \text{pow}\ (2, \log(2)))) \tag{2}$$

$$p = \text{pow}\ (1 - \exp (-k / (m / n)), k) \tag{3}$$

$$n=\text{ceil}\ (m/\log(1-\exp(\log(p)/k))) \tag{4}$$

IV. Basic Design of the Mechanism

All the route nodes that lie in the band are assumed to be friends because the source has contributed its authentication key with all those nodes. The information M is shared through the routing path $R_T= \{R_1, R_2, \ldots, R_n\}$ to the target node T by the source S. Initially, a fundamental system with an issue is presented, and then we propose a further greater creation is shown inside the subsequent phase. To simplify this, we pass over by using what manner the networks come to a selection and provide routing paths.

A. *Basic Technique*

The construction of the MAC manifest using the Bloom filter is considered the entire working system for step-by-step recognition. En-route authentication with a map is the output obtained by applying the above technique. While other methods use a particular receiver to evaluate the genuineness of information en route authentication, the bitmap uses one or more receivers. En-route authentication code is much smaller than the message authentication code. Thus, EAB utilizes a lower amount of bandwidth for transmitting. Hence it becomes fitter to travel through the routing path. When an en-route authentication code is employed, every router in between is insisted to allow only false information of less liability. Hence, it becomes compulsory for the foe to breach an authorized EAB to impart a false message to a node where each router allows only less liable false messages.

B. *Protocol Description*

1) Source node: Firstly, the validation parameters are determined by the source node based on its privacy needs. There begins the multi-hop routing along S, R, and T. The source node initially creates a signature that is responsible for the start of end-to-end authentication. To generate the signature, either message authentication uses a public key code. Only the source node has the message authentication code key. There is no threat to the security by the adversary since M is in the encrypted state. Here the message by S is M' ← {S |T | M |sig}.

Every route H_i gets validated by S with the help of an en-route authentication bitmap. Using this source node pare wise key, the message authentication code is generated by Eq. 5.

$$EAB[x] = \begin{cases} 1 & \exists_i \exists_j, x = H_j(MAC(KS, R_i, M')) \\ 0 & otherwise \end{cases} \quad (5)$$

When including H_i, the message authentication code is interlinked with the m-bit filter; hence, the en-route authentication code is generated when the particular message, as well as its path, is concerned. At last, the initial root node receives the combination of information and its en-route authentication code. Algorithm 1 explains the source node procedure.

2) Destination node: The best authentication method for sigS or TM is to authenticate M at the destination.

3) Route node: Ri gets a message with the EAB BFS, R(M') for each routing node, which makes certain of the following situations. The number of marked bits in EAB falls within the variety according to Eq. 6.

$$[\lfloor mp - \delta \rfloor, \lceil mp + \delta \rceil] \quad (6)$$

$$\delta = \sqrt{mp(1-p)}$$

$$p = 1 - (1 - 1/m)\ k_n$$

We will make clear the effect of δ in the element deferred. If each situation is contented, Ri will course the message at the side of the filter out to the subsequent router Ri+1. Afterward, it separates the message and reports some false facts. Algorithm 2 describes the procedure in detail.

For instance, as shown in Fig. 4, the source node shapes an augmented Bloom filter of size m = 32 for the diffusion over N = 8 path nodes. Every MAC (KS, Ri, M') maps to k = three marked bits in EAB. The ultimate EAB is the bit-wise AND of all plotting. On routine, the EAB will incorporate about 15 marked bits. According to Fig. 5 and 6, the mapped bits in EAB are segmented by MAC.

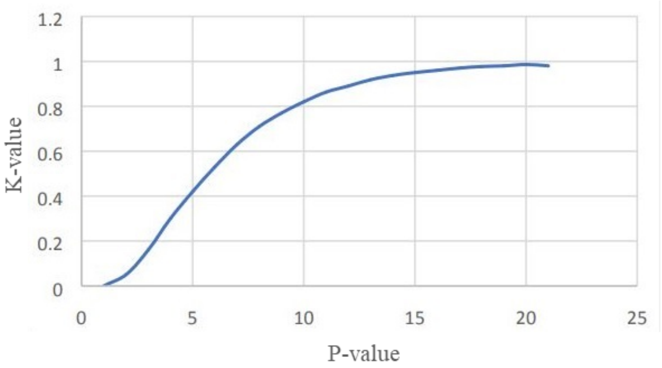

Fig. 4. Graph p vs. k.

C. Accomplishment of Hash Set

Generally, it is not necessary for S to implement a new hash set h to work on every message authentication code. In case cryptographic hash functions are used in message authentication code, the marked bits are automatically defined by the output gained. When the image is fragmented to 160/5, that is 32 pieces. The initial three pieces are selected by S as a reference to the market bits in the En authentication code. It greatly adds an advantage when the cost is concerned, where the root node must implement only one hash function to validate the information.

D. Security Analysis

As the message authentication codes are directly involved in the development of en-route authentication code, the MAC keys are confidential; the breach system mainly aims to break through en-route authentication, aiming its false messages, where breaking in maximum probable route nodes is planned. Here, en-route authentication consists of hash functions generated by the message authentication code of various route nodes, in which the attacker has individual distributions. So, ignoring any node on the way could cause the attacker's intentions to fail.

1) Probability of bypass: With the usage of a 32-bit en-route authentication bitmap and k=3 hash by the source node, the attack of the adversary reduces up to 81 percent before the completion of two steps. Security increases as the size of the en-route authentication bit increase, but the bandwidth also increases. This attack probability further decreases when 64-bit Android authentication bitmap and k=6 hash function. The adversary must inject a maximum of 1400 information to breach at least the third step node even with half chances. The probability is given by Eq. 7.

$$P_{bypass} = (1 - (\frac{1}{m})^{nk})^t \quad (7)$$

2) The parameter k: To calibrate the bloom filter, the amount of hash function is to be calibrated. If k=ln 2m/n, the en-route authentication bitmap would contain the maximum false positive.

The parameter δ: The attacker can produce fewer Mark bits by optimizing this parameter. This makes the en-route authentication bitmap to be a binomial distribution. It is not necessary e for the mark bit to be *mp* always. The common number of marked bits can be between $\delta = 2\sqrt{mp(1-p)}$ whose chances is higher than 90%. It would be very easy for the attacker to break all the barriers easily without this parameter. There are some disadvantages. The attacker sets the largest possible number of Marked bits instead of the optimum amount to increase the winning chances. In some cases, S would create an en-route authentication map.

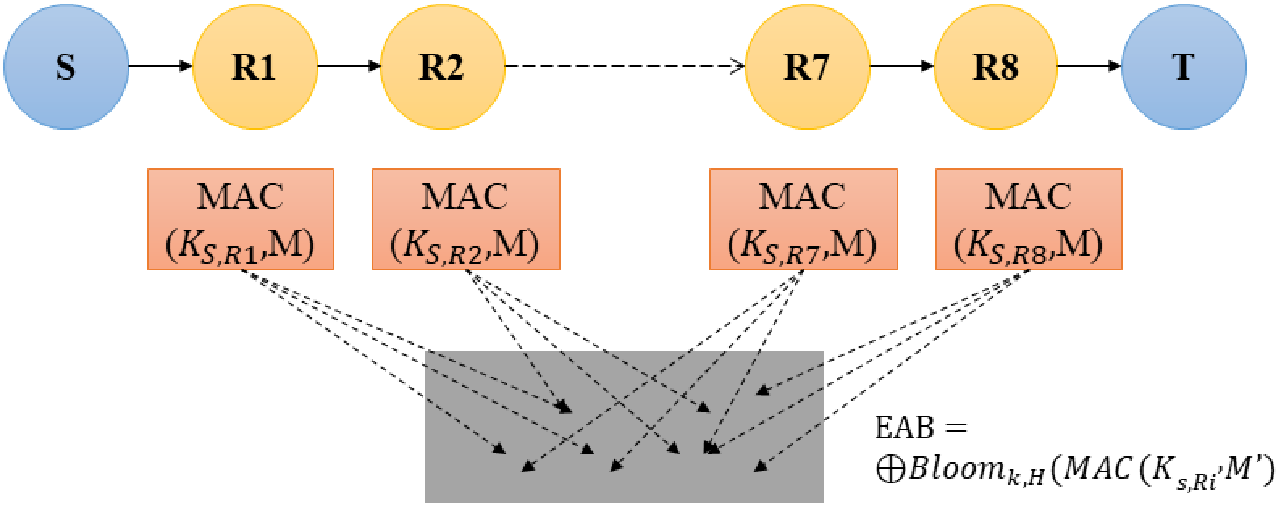

Fig. 5. En-route authentication bitmap.

Algorithm 1. Source node procedure

```
compute M' ← {S |T | M |$sig_{S,T}$ (M)}
clean EAB
for i=1 to n do
    $Bitmap \xleftarrow{bloom} H, MAC(K_{S,R_i}$, M')
    EAB $\bigoplus bitmap_i$
end for
Send out { M'|EAB}
```

Algorithm 2. Route node R_i procedure

```
Receive packet { M'|EAB} from router $R_{i-1}$
If ($\lfloor mp - \delta \rfloor \leq count\ (EAB) \leq \lceil mp + \delta \rceil$) then
    for j = 1 to k do
        compute $x_j' \leftarrow H_j(MAC_{S,R_i}$(M'))
        if (EAB [$x_j'$] = 0) then
            drop the packet
        end if
    end for
    forward the packet to the next route $R_{i+1}$
else
    drop the packet
end if
return
```

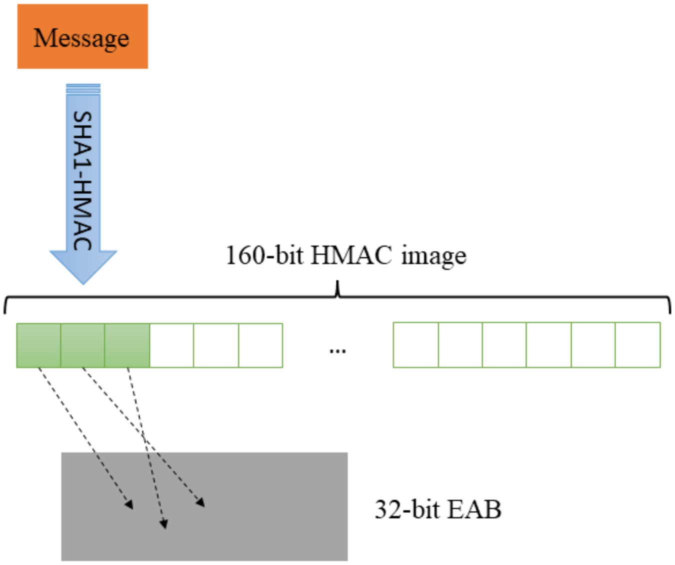

Fig. 6. The segmentation of MAC.

V. Improvement with Random Padding

We have proposed an easy remedy for this problem. This can increase privacy to the next level. The information is as follows when source node S is supposed to possess two pots.

$$M' \leftarrow \{S|T|M|\text{ padding }|sig_{S,T}(M|padding)\} \quad (8)$$

Less size of stochastic string that is chosen by S is said to be a wedding. Hence, the en-route authentication bitmap depends on the number of marked bits. Till the information m gives an output en-route authentication bitmap with marked bits less than β, the source node sets the padding stochastically until the message is sent. The value of beta is a common and accepted value by each routing node. This helps S to check different en-route authentication bitmaps for constant information. Hence, it would be easy for the source node to discard insecure en-route authentication bitmaps. However, this checking time method should be reduced if successful results come in series. There is no need for the padding to be larger to succeed in every aspect. In line 2 of algorithm 2, S changes its characteristics as follows.

$$count\ (EAB) \leq \beta \quad (9)$$

The approximate number of times of testing the good en-route authentication bitmap as Eq. 8 is as follows.

$$\frac{1}{\sum_{i=1}^{\beta}\binom{m}{i}p^{i}(1-p)^{m-i}} \quad (9)$$

For example, according to Fig. 7, when *M* is 32, *n* is 8, and *k* is 2, the source node generates an en-route authentication bitmap with marked bits with bits below 13, and this can reduce the injection of false data up to 94.73% in the first and second steps it further gets reduced to 99. 72%. When a 4-bit is padded, this probability goes down to $3.8*10^{21}$.

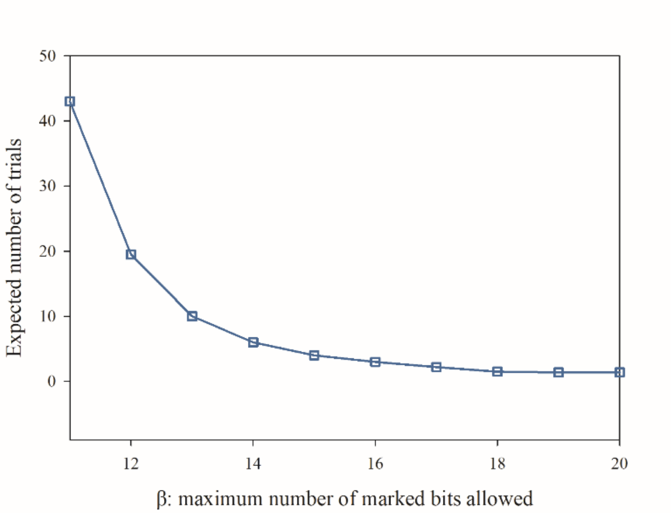

Fig. 7. The expected number of trials to find the desired EAB.

As shown in Fig. 8, for Bloom filters with m = 32 and n = 8, the optimal k value should be 3, which will increase the probability of passing through route nodes. The well-selected EAB of β = 15 under k = 4 may have superior performance compared to smaller β under smaller k. Nevertheless, more padding tests or even greater padding will be needed if the number of hashes increases. According to Eq. 10, the parameters depend on the network requirements.

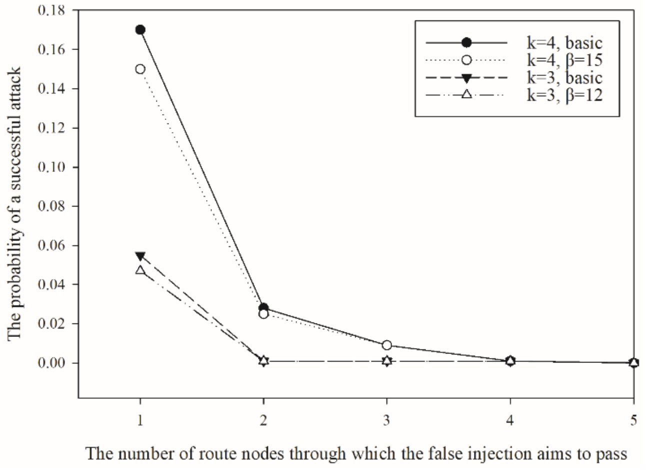

Fig. 8. The probability of bypass in fine-tuned EAB.

$$\frac{1}{\sqrt[n]{S}} \leq \left(\frac{\beta}{m}\right)^{k} \leq P_b \quad (10)$$

VI. Conclusion

This paper proposed a lightweight answer for step-by-step authentication in IoT. Low computational cost, high accuracy, and low communication overhead characterize our shape. Authentication is performed by combining Augmented Bloom filters with bitmaps from en-route authentication. The Augmented bloom filter utilizes a single hash function named Message Digest (MD5). Its main intention is to thoroughly remove the false negative rate and highly lessen the false positive rate. Single hash function usage results from more false positive rates, but we impulse the MD5 here. We split the input into three and applied the MD5 hash function to each of them separately to collect three different key values. This idea works great on it and successfully results in a less false positive rate. The most important goal of EAB is to categorize false records inside the first limited steps as a substitute for dropping them at the beginning sign. Thus, we admit an adequate false-negative rate adjoining the adversary, but the false-negative rates quickly come together as the next count increases. This interchange significantly diminishes the communique overhead received through authentication. We declare that such remedy can be retained similarly to light-weight authentication outlines in cooperative environments.

REFERENCES

[1] Mousavi, S.K., et al., Security of internet of things based on cryptographic algorithms: a survey. Wireless Networks, 2021. 27(2): p. 1515-1555.

[2] Ataie, I., et al. D 2 FO: Distributed Dynamic Offloading Mechanism for Time-Sensitive Tasks in Fog-Cloud IoT-based Systems. in 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC). 2022. IEEE.

[3] Seyfollahi, A., T. Taami, and A. Ghaffari, Towards developing a machine learning-metaheuristic-enhanced energy-sensitive routing framework for the internet of things. Microprocessors and Microsystems, 2023. 96: p. 104747.

[4] Mehbodniya, A., et al., Energy-Aware Routing Protocol with Fuzzy Logic in Industrial Internet of Things with Blockchain Technology. Wireless Communications and Mobile Computing, 2022. 2022.

[5] Sellami, B., et al., Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network. Computer Networks, 2022. 210: p. 108957.

[6] Haghshenas, S.H., M.A. Hasnat, and M. Naeini, A Temporal Graph Neural Network for Cyber Attack Detection and Localization in Smart Grids. arXiv preprint arXiv:2212.03390, 2022.

[7] Mishra, S. and A.K. Tyagi, The role of machine learning techniques in internet of things-based cloud applications, in Artificial Intelligence-based Internet of Things Systems. 2022, Springer. p. 105-135.

[8] Cauteruccio, F., et al., A framework for anomaly detection and classification in Multiple IoT scenarios. Future Generation Computer Systems, 2021. 114: p. 322-335.

[9] Ferrag, M.A. and L. Shu, The performance evaluation of blockchain-based security and privacy systems for the Internet of Things: A tutorial. IEEE Internet of Things Journal, 2021. 8(24): p. 17236-17260.

[10] Ren, P., et al., IPSadas: identity ‐ privacy ‐ aware secure and anonymous data aggregation scheme. International Journal of Intelligent Systems, 2022. 37(8): p. 5290-5324.

[11] Shah, P. and T. Kasbe, A review on specification evaluation of broadcasting routing protocols in VANET. Computer Science Review, 2021. 41: p. 100418.

[12] Montanari, A.N., et al., Functional observability and target state estimation in large-scale networks. Proceedings of the National Academy of Sciences, 2022. 119(1): p. e2113750119.

[13] Pourghebleh, B., et al., A roadmap towards energy ‐ efficient data fusion methods in the Internet of Things. Concurrency and Computation: Practice and Experience, 2022: p. e6959.

[14] Mohseni, M., F. Amirghafouri, and B. Pourghebleh, CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic. Peer-to-Peer Networking and Applications, 2022: p. 1-21.

[15] Saeidi, S.A., et al. A novel neuromorphic processors realization of spiking deep reinforcement learning for portfolio management. in 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2022. IEEE.

[16] Zheng, Y., et al., PUF-based Mutual Authentication and Key Exchange Protocol for Peer-to-Peer IoT Applications. IEEE Transactions on Dependable and Secure Computing, 2022.

[17] Chen, X., et al. A blockchain based access authentication scheme of energy internet. in 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2). 2018. IEEE.

[18] Lawrence, T., et al., A computationally efficient HMAC-based authentication scheme for network coding. Telecommunication Systems, 2022. 79(1): p. 47-69.

[19] Nianmin, Y., M. Haifeng, and H. Yong, A method for memory integrity authentication based on bloom filter. Journal of Algorithms & Computational Technology, 2014. 8(3): p. 267-286.

[20] Saravanan, K. and A. Senthilkumar, Security enhancement in distributed networks using link-based mapping scheme for network intrusion detection with enhanced Bloom filter. Wireless Personal Communications, 2015. 84(2): p. 821-839.

[21] Wang, J., et al. An authentication key agreement scheme for heterogeneous sensor network based on improved counting bloom filter. in 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC). 2015. IEEE.

[22] Malhi, A. and S. Batra, Privacy-preserving authentication framework using bloom filter for secure vehicular communications. International Journal of Information Security, 2016. 15(4): p. 433-453.

[23] Mbarek, B., N. Sahli, and N. Jabeur. BFAN: A Bloom Filter-Based Authentication in Wireless Sensor Networks. in 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC). 2018. IEEE.

[24] Sateesan, A., et al. Novel Bloom filter algorithms and architectures for ultra-high-speed network security applications. in 2020 23rd Euromicro Conference on Digital System Design (DSD). 2020. IEEE.

[25] Borgia, E., The Internet of Things vision: Key features, applications and open issues. Computer Communications, 2014. 54: p. 1-31.

[26] Zhu, S., et al. LHAP: a lightweight hop-by-hop authentication protocol for ad-hoc networks. in 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings. 2003. IEEE.

[27] Akbani, R., T. Korkmaz, and G. Raju. HEAP: Hop-by-hop efficient authentication protocol for mobile ad-hoc networks. in Proceedings of the 2007 spring simulaiton multiconference-Volume 1. 2007.

[28] Heer, T., et al. Alpha: an adaptive and lightweight protocol for hop-by-hop authentication. in Proceedings of the 2008 ACM CoNEXT Conference. 2008.

[29] Chen, Y.-S., et al. Broadcast authentication in sensor networks using compressed bloom filters. in International Conference on Distributed Computing in Sensor Systems. 2008. Springer.