applied sciences

MDPI

Article

A Secure Communication Method Based on Message Hash Chain

Mingxuan Han and Wenbao Jiang *

Department of Information Security, Beijing Information Science and Technology University, Beijing 100192, China; hanmingxuan@bistu.edu.cn
* Correspondence: jiangwenbao@tsinghua.org.cn

Abstract: Traditional network communication methods lack endogenous security mechanisms, which is the root cause of network security problems, e.g., spoofing identity and address forgery. This paper proposes a secure communication method based on the message hash chain, referred to as the chain communication method or MHC method. We use the message hash chain to ensure that the transmission process is immutable, non-repudiation, reliability, and the integrity and synchronization of the message. At the same time, we can sign and authenticate data streams in batches through chain signature and authentication technology, which can significantly reduce the overhead of signature and authentication, thereby improving the efficiency of secure message transmission. This paper formally proves the security of the message hash chain, conducts an in-depth analysis of the reliability of the MHC method, and conducts relevant experimental tests. The results show that the average transmission efficiency of the MHC method applied at the network layer is about 70% lower than that of the IP protocol communication method without a security mechanism. However, it is about 5% higher than the average transmission efficiency of the non-repudiation IPSec protocol communication method. The average transmission efficiency of the MHC method is about 23.5 times higher than that of the IP protocol communication method with the packet-by-packet signature. It is easier to ensure the non-repudiation of the data stream.

Keywords: message hash chain; chain communication method; chain signature; endogenous safety

Citation: Han, M.; Jiang, W. A Secure Communication Method Based on Message Hash Chain. *Appl. Sci.* **2022**, *12*, 4505. https://doi.org/10.3390/app12094505

Academic Editors: Leandros Maglaras, Helge Janicke and Mohamed Amine Ferrag

Received: 22 March 2022
Accepted: 21 April 2022
Published: 29 April 2022

1. Introduction

Nowadays, network communication applications are ubiquitous, causing various security problems. The data receiver wants to get all the data content sent by the sender and wants the data to be complete, authentic, and non-repudiation. At the beginning of the design of the existing network communication methods, the focus is only on data transmission connectivity, while data transmission security is ignored. This design fundamentally lacks endogenous security mechanisms and is also the root cause of security problems such as identity spoofing, address forgery, route hijacking, and denial of service in cyberspace. Moreover, the weak association between each message in the data stream leads to low reliability of the transmission process.

Traditional network communication methods do not have endogenous security mechanisms such as data integrity verification, making the transmitted content easy to be tampered with and forged, making it difficult to trace the source of the attack and the attacker's identity. To solve such security problems, the IPSec [1] (IP Security) security suite of the network layer is mainly used to perform integrity verification, data encryption, and data source authentication on the transmitted IP datagrams. Nevertheless, IPSec can usually only solve local problems on a regional scale. In particular, implementing IPSec technology is relatively complex, requiring two stages of negotiation before data transmission. The time and computing resources consumed by each step of the negotiation process are rather significant [2], necessarily leading to the problem of poor deployability.

Reference [3] has proposed an attack method for the first-phase authentication process, and the IKE protocol used in the negotiation also has vulnerabilities such as man-in-the-middle attack [2] denial of service attack [4]. At the same time, the authentication header protocol [5] (Authentication header, AH) and the encapsulating security payload protocol [6] (Encapsulating Security Payload, ESP) are included in IPSec. However, the AH protocol can ensure the integrity of the transmitted messages, data source authentication, and anti-replay protection services. The ESP protocol can also provide data stream encryption services. Both protocols can easily guarantee the non-repudiation of the message, and both communicating parties can effectively synchronize the message and trace the message.

The latest technologies in vehicular ad hoc networks and the Internet of Things (IoT) provide solutions to traditional networks that lack security and trust mechanisms [7–9]. These technologies ensure the authenticity, reliability of the information in the network, and the legitimacy of the vehicles disseminating such information. In traditional networks, the authenticity and reliability of packets transmitted between network nodes and the trust between nodes are also crucial. We consider that constructing a "chain" of messages communicated between nodes in a traditional network can provide a secure and reliable mechanism for the network. Lamport first proposed the concept of a "hash chain" to solve the problem of password tampering during transmission [10]. Existing research on the hash chain only constructs various forms of hash chain structures for application-layer data. These studies make hash chains computationally expensive for security reasons. However, none of these schemes use a sequence of network communication messages to construct a hash chain nor a synchronization mechanism for network communication messages. At the same time, these schemes all use a hash chain to encrypt data or keys to achieve higher security for data content while preventing encrypted content from being cracked and tampered with, and none of the solutions is to improve the efficiency of secure data transmission.

Contributions

Aiming at the shortcomings of the above traditional network communication methods, we propose a novel secure communication method based on the message hash chain, referred to as the Message Hash Chain (MHC) method. The main contributions of the proposed MHC method are summarized as follows:

1. The MHC method adopts a new chain transmission method to ensure the non-tampering, non-repudiation, and higher reliability requirements of multiple messages. The main idea is to iteratively hash the digest of the transmitted message to form a hash chain about the message sequence. The two communicating parties can ensure the integrity, immutability, and synchronization of the message sequence through the hash chain, thereby effectively guaranteeing the security of message transmission.
2. When performing data signature and authentication, both parties only need to perform signature authentication on messages at certain intervals and do not need to complete it on each message. In this way, the authenticity and non-repudiation of all previously transmitted messages can be ensured, the overhead of signature authentication is reduced, and the efficiency of secure message transmission is greatly improved.
3. Using the sequence number and node value of the message hash chain of the MHC method can provide anti-protection against replays and ensure reliability.

2. Related Works

The method proposed by Lamport is to encrypt the password through the hash function many times iteratively, and the verifier can verify the entire ciphertext sequence through the result of the latest encryption.

Based on Lamport, Chung et al. [11] proposed the star chaining technique and tree chaining technique. The star chaining technique can verify each packet individually and can tolerate any degree of packet loss. The tree chaining technique can be regarded as a multi-layer star chaining technique. Although this scheme can achieve a smaller communication

load than a star hash chain, it disadvantages sender delay, buffering of packets before sending, and less payload.

Golle [12] proposes a hash chain with high performance and a high proportion of payload, but its biggest flaw is that it cannot avoid the risk of chain disconnection caused by too many packets contained in the chain.

Liu [13] proposed a hash pre-streaming data signature scheme. The basic idea is to divide a long sequence into m subsequences and use the hash pre-streaming data signature scheme to sign the first packet of the m subsequences. At the same time, a buffer dedicated to storing the hash values and signatures of the n packets in the subsequence is added to the server.

Zhang et al. [14] proposed a butterfly-graph-based stream authentication scheme with advantages in payload, packet authentication probability, and packet loss tolerance. However, compared with other structures of hash chains, this method needs to run the hash function many times, making it less efficient.

Miller et al. [15] improved the scheme proposed by Zhang. Although the security of the hash chain and the probability of data packet authentication were strengthened on the original basis, the complex structure led to a further decrease in its operating efficiency.

The authentication protocol based on hash chain proposed by Liu [16] can calculate a continuous hash chain by performing multiple hash function calculations on the hash value of the data payload. Although the biggest feature of this authentication protocol is that it can resist replay attacks, it still cannot guarantee the non-repudiation of each packet.

Huang et al. [17] used different hash functions to iterate keys multiple times and finally got a hash chain authentication scheme for message integrity verification. Still, this scheme's order of hash functions needs to be kept secret.

References [18,19] propose self-updating hash chains and optimized tree hashing, respectively. These two hash chain structures optimize the security and packet loss tolerance on the original basis. Still, the overall operating efficiency is not much different or even slightly insufficient from the original structure.

The concept of "hash chain" is currently widely studied in application fields such as the Internet of Things, autonomous driving protocols, data security, and lightweight transmission protocols. Hakeem et al. proposed a hash chain-based V2X security protocol and a key generation and management protocol at [20,21]. The primary method uses the hash function to iterate the generated key many times, which realizes the highly secure message authentication in the V2X device at a low cost. At the same time, it can solve the key update problem of remote WAN and can resist key leakage attacks and replay attacks. Huang et al. [22] proposed a hash chain-based data availability monitoring method, which applies the hash chain to the distributed system to solve the data consistency problem in the system. Kim et al. [23] proposed a lightweight authentication scheme applied to military networks. This scheme combines the hash chain with the one-time password, which ensures the integrity of the transmission content and reduces the network transactions of transmission. Luo et al. [24] improved the blockchain consensus algorithm by using the hash chain to realize the recording and verification of blocks.

3. Chain Communication Model

3.1. Notation and Meaning

The notation involved in this paper and their corresponding meanings are shown in Table 1.

3.2. Message Hash Chain Communication Model

The MHC method is not only for a specific layer in the TCP/IP network model but also for each message in the data flow, which can be applied to any logical layer. The structure diagram of the chain communication model is shown in Figure 1. The MHC method adds the message sequence number (Sequence) and the node value of the message hash chain fields. The sequence is used to provide the reliability of the transmission process, and the

node value of the message hash chain is used to verify the message. Its construction process is described in detail in Section 4.

Table 1. Notation and meaning.

Notation	Meaning
$h(\cdot)$	Cryptographic hash functions, $h : \{0,1\}^* \rightarrow Z_q^*$.
$A\|\|B$	Concatenate string A with string B.
p_i	The ith message of message transmission sequence.
HC_i	The ith node value of the message hash chain constructed by the sender is sent to the communication peer together with the payload and needs to be verified.
HC_i'	The ith node value of the message hash chain constructed by the receiver is used to compare with the received message hash chain.
m_i	The content of the ith message sent by the sender.
s_i	The digital signature of p_i.
(ek, dk)	Key pair in the asymmetric digital signature.
$Sig(ek, HC_i)$	Based on its ek, the sender uses an asymmetric encryption algorithm to calculate the signature s_i of HC_i.
$Ver(dk, s_i)$	The receiver verifies the signature s_i against the sender's $publickey$. If the value is 1, it means that the verification can be successful; otherwise, it means that the verification cannot be passed.

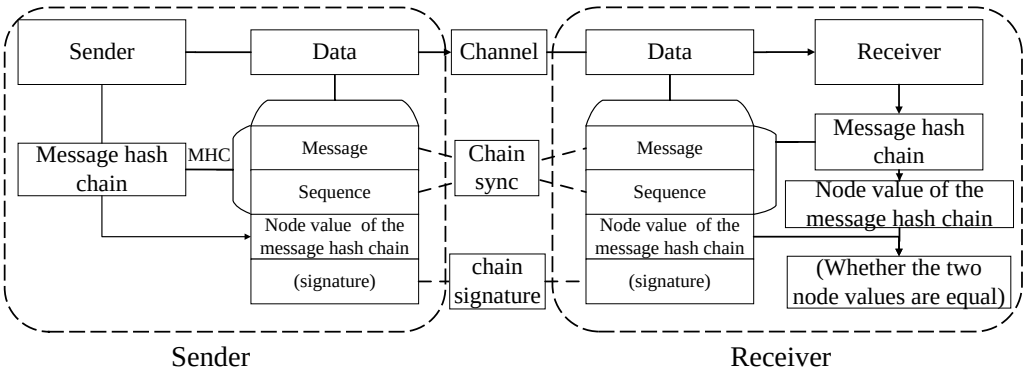

Figure 1. Message hash chain communication model.

The chain communication model mainly includes the sender constructing the message hash chain and the receiver verifying the message hash chain during the interaction between the two communicating parties. The specific processing procedures of the sender and receiver are as follows.

1. The sender first determines the src(source address), dst(destination address), and $other$(other fields of the header) of the sent message, and then the header information $hdr_i = (src, dst, other)$ can be obtained. The expression of the message m_i that the sender needs to send in the MHC method is

$$m_i = (hdr_i, payload_i). \quad (1)$$

where $payload_i$ is the payload of m_i. The sender needs to obtain the message hash chain from src to dst locally and record the message hash chain as $HC(src, dst)$. According to the m_i and the tail node $HC(src, dst)_{i-1}$ of the message hash chain, the latest node $HC(src, dst)_{i-1}$ of the message hash chain is constructed, i.e., the node value $HC(src, dst)_i = h(h(m_i)||HC(src, dst)_i - 1)$ of message hash chain corresponding to the message m_i. At the same time, HC_{i-1} is updated to the intermediate node of this message hash chain, and HC_i is updated to the tail node of the chain of this message hash chain. The sender sends the message $p_i = (m_i, seq_i, HC_i)$ to the receiver according to dst, where seq_i is the sequence.

2. After receiving the p_i, the receiver verifies the node value of the message hash chain. Record the receiver's message hash chain as $HC'(src, dst)$. The receiver calculates a node value $HC'(src, dst)_i = h(h(m_i)||HC'(src, dst)_{i-1})$ to be verified in the message hash chain between *src* and *dst* through the construction method of the message hash chain, where $HC'(src, dst)_{i-1}$ is the tail node of the message hash chain constructed by the receiver. Next, the receiver verifies whether $HC(src, dst)_i = HC'(src, dst)_i$ holds. If so, the The receiver's verification of p_i ends successfully and updates $HC'(src, dst)_{i-1}$ to the intermediate node of the message hash chain and $HC'(src, dst)_i$ to the tail node of the message hash chain. Otherwise, the receiver's verification of p_i is unsuccessful and must to discard the p_i and report an error.

3.3. Message Structure

The MHC method forms a message hash chain from unrelated data packets. The receiver can use the node value of the message hash chain to verify the integrity of the current message content and ensure the immutability of the data stream. When the receiver is affirming the packet, the verification succeeds only if the value calculated by using the digest of the previous message and the node value of the message hash chain is equal to the node value of the chain carried in the message. The node value of the chain corresponding to each message in the data flow constitutes a message hash chain, as shown in Figure 2. Through this message structure, the integrity of the message can be verified, but the reliability of the transmission process can be improved.

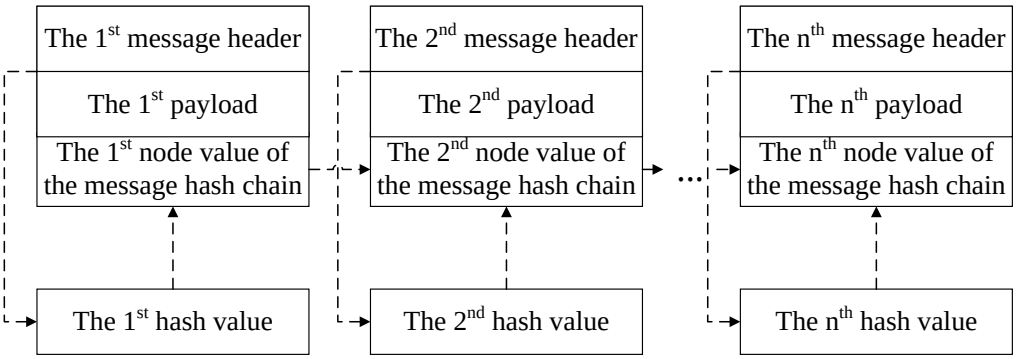

Figure 2. Schematic diagram of message hash chain structure.

4. Construction Method of Message Hash Chain

Two communicating parties, A and B, communicate, and sender A transmits the message stream $P = p_1, p_2, \cdots, p_n, n \in N^*$ to receiver B. The structure of each message is $p_i = (m_i, seq_i, HC_i)$. Where m_i is the content of the message sent by sender A in sequence, seq_i is the sequence number of the message, and HC_i is the result calculated by sender A according to m_i and the tail node HC_{i-1} of the message hash chain by the constructing method of the chain. When receiver B receives p_i, it also needs to use m_i and the local tail node HC'_{i-1} of the message hash chain to calculate the HC'_i for verification.

The construction method of the message hash chain is shown in Figure 3. The communication node needs to calculate the first node value HC_1 of the message hash chain according to the first message m_1, obtained by performing two hash function calculations on m_1. After that, each message needs to calculate a digest using a hash function and then splice this digest with the tail node of the message hash chain to calculate the corresponding node value of the chain.

The last node of each message hash chain is called the tail node, and the other nodes are called the intermediate nodes. The sender updates the node of the message hash chain corresponding to the latest sequentially sent message to a tail node and updates the original tail node to an intermediate node. The receiver verifies the messages in sequence and uses the successfully verified messages to construct a node of the message hash chain. Update this newly constructed node to the tail and the previous tail node to the intermediate nodes.

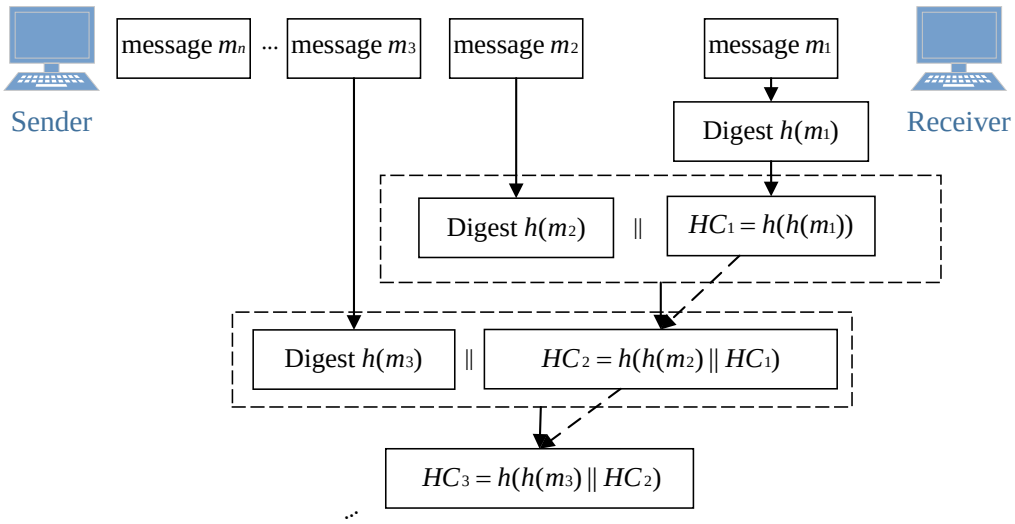

Figure 3. Construction method of the message hash chain.

The iterative process of the message hash chain node is shown in Algorithm 1. The parameters used in Algorithm 1 are described below:

- *Get_address*: The role of the *Get_address* function is to obtain the source and destination addresses from the message header.
- *Match_HC_lnode*: Match the message hash chain between two addresses.

Algorithm 1 *HC_Iteration*

Input: Header content, payload, node value of the message hash chain.
Output: A new node value of the chain.

1: $pkt\_hash \leftarrow Hash(hdr, seq, payload)$
2: $src, dst \leftarrow Get\_address(hdr)$.
3: $HC\_lnode \leftarrow Match\_HC\_lnode(src, dst)$.
4: **return** $Hash(HC\_lnode, pkt\_hash)$

The message hash chain construction algorithm is shown in Algorithm 2.

Algorithm 2 The construction process of the message hash chain

1: $HC\_lnode =$ ""
2: **for** *MQueue* is not empty **do**
3: $payload \leftarrow MQueue.poll$
4: $HC\_lnode \leftarrow HC\_Iteration(hdr, payload, HC\_lnode)$
5: **end for**

The two communicating parties update the message hash chain every time they construct a message hash chain node. At a specific time t, a message hash chain node of m_i is constructed, then the complete message hash chain expression at time t is as following:

$$HC_i = \begin{cases} h(h(m_i)||HC_{i-1}), & i \geq 2 \\ HC_1, & i = 1 \end{cases}. \quad (2)$$

5. Chain Synchronization

5.1. Sequence Number

The MHC method needs to add a sequence field to ensure the reliability of the transmission process. According to the position of the sequence number, when the node value of the message hash chain is calculated, there are two ways to calculate the chain. The first

way is that the sequence number seq_i can be included in the message header hdr_i, and the node value HC_i of message hash chain is obtained by MHC calculation. In this way, the non-repudiation of the sequence field can be guaranteed, but it will cause difficulties when tracing the message's contents. The message hash chain expression constructed in this way is the following:

$$HC_i = h(h(m_i||seq_i)||HC_{i-1}). \tag{3}$$

The second way is to concatenate the digest of the sequence number and the message's digest to construct the message's node value of the message hash chain. This way ensures that the message and sequence cannot be tampered with, and makes it easier to trace the message's content. Therefore, it is recommended to use the second way when constructing the node value of the message hash chain. The expression of the node value of the message hash chain constructed in this way is the following:

$$HC_i = h(h(m_i)||h(seq_i)||HC_{i-1}). \tag{4}$$

5.2. Chain Synchronization Mechanism

5.2.1. General Chain Synchronization Mechanism

In order to solve the problem of locating the error and re-request and verifying the message when the message hash chain verification or signature verification fails in the MHC method, we propose a communication synchronization mechanism, referred to as the chain synchronization mechanism. The MHC method uses this mechanism to maintain the consistency of the message hash chain of both parties. Two communicating nodes are communicating via the MHC method, A sending data stream $P = p_1, p_2, \ldots, p_n$ to B. Note that the message hash chain constructed by the sender is HC, and the chain constructed by the receiver is HC'. Whenever a message $p_\delta = (m_\delta, seq_\delta, HC_\delta)$ satisfy $HC_\delta = h(h(m_\delta)||HC'(\delta - 1))$ during verification, its content is wrong, and the receiver needs to re-request this message from the sender. On the contrary, the receiver can successfully verify the p_δ and continue to verify $p_{\delta+1}$.

5.2.2. Chain Synchronization Mechanism Based on Signature Confirmation

Assuming that the interval between the chain signatures of two communication nodes is d, the sequence numbers seq_α and seq_β corresponding to the two chain signatures s_α and s_β should satisfy $\beta = \alpha + d$. If $\exists\delta, \alpha < \delta < \beta$, starting from p_δ, the attacker can use the algorithm $Attack$ to tamper with the content of the message and make it successfully pass the receiver's message hash chain verification (it is challenging for the attacker to do this). Subsequently, the message tampered with by algorithm $Attack$ is recorded as $p^*_\delta = (m^*_\delta, HC^*_\delta)$, and the message hash chain constructed by the receiver according to $HC^*_\delta, HC^*_{\delta+1}, \ldots, HC^*_{\beta-1}$ is recorded as $HC'(Attack)$. When the sender reaches the signature interval (or actively signs the chain as needed), the receiver must verify $p_\beta = (m_\beta, HC_\beta, s_\beta)$, satisfying as the following:

$$\begin{cases} Ver(dk, s_\beta) = 1 \\ h(h(m_\beta)||HC'(Attack)_{\beta-1}) \neq HC_\beta \end{cases}. \tag{5}$$

Then the receiver needs to re-request $p_{\alpha+1}, p_{\alpha+2}, \ldots, p_{\beta-1}$, and the receiver's message hash chain needs to be reconstructed from HC'_α.

6. Chain Signature

We improved the chain signature scheme previously proposed in [25] to achieve higher security and efficiency. By Section 7, the (Gen, Sig, Ver) scheme is an additional option of the (MHC, Gen, Sig, Ver) scheme, enabling the MHC method to guarantee the authenticity and non-repudiation of data. In this way, the (MHC, Gen, Sig, Ver) scheme can verify all previous messages with only one signature, dramatically improving signature and authentication efficiency. Suppose there is a message $m_\delta = (hdr_\delta, payload_\delta)$, and its

corresponding message hash chain node at the sender is HC_δ. If the sender reaches the signature interval or chain-signatures the message as required, the signature $s_\delta = Sig(ek, HC_\delta)$ must be calculated first, and then the encapsulated message $p_\delta = (m_\delta, HC_\delta, s_\delta)$ is sent to the receiver. Suppose the receiver can successfully verify the node value and signature of the message hash chain in the p_δ in turn, i.e., in that case, the receiver can satisfy the equations $HC_\delta = HC'_\delta$ and $Ver(dk, s_\delta)$ when verifying the p_δ, and it can guarantee the non-repudiation of all previously transmitted messages.

6.1. Chain Signature Process

Algorithm 3 shows the process of chain signature for both parties in communication. The messages transmitted by the two communicating parties include messages with a signature and those without a signature, and the chain signature interval is d. In the process, the communication node constructs the message hash chain and transmits the messages synchronously, e.g., the node encapsulates the m_δ and HC_δ constructed according to the m_δ into a message p_δ and sends it to the destination. For the security of the message hash chain, the sender will chain-sign the message when the signature counter reaches d or when necessary, e.g., after the sender signs HC_δ, it only needs to sign $HC_{\delta+d}$ next time. The structure of a message with a signature is $p_i = (m_i, seq_i, HC_i, s_i)$, and a message without a signature is $p_j = (m_j, seq_j, HC_j)$. The process of the sender encapsulating the messages shown in Algorithm 3. The parameters used in Algorithm 3 are described below:

- *cur_interval*: Current signature interval.
- *Sig_interval*: A signature is required when the sender's signature interval reaches *Sig_interval*.
- *EnPkt*: The function that encapsulates parameters as header of message hash chain.
- *Sig*: The signature function described in Section 7.
- ek_{src}: Sender's private key.

Algorithm 3 Message Hash Chain Encapsulates Messages Header

Input: Header content, signature interval, payload.
Output: Encapsulated MHC datagram.

1: According to the content of the message, the payload and the tail node of the message hash chain, a node value $HC\_node \leftarrow HC\_Iteration(hdr, payload, HC\_lnode)$ of the chain is generated.
2: The sender inserts *HC_node* at the end of the message hash chain.
3: The sender updates message hash chain tail node $HC\_lnode = HC\_node$.
4: **if** $cur\_interval < Sig\_interval$ **then**
5: The sender encapsulates the header $p \leftarrow EnPkt(hdr, payload, HC\_node)$.
6: **else**
7: The sender computes the signature $s \leftarrow Sig(ek_{src}, HC\_node)$.
8: $p \leftarrow EnPkt(hdr, payload, HC\_node, s)$
9: **end if**
10: **return** p

6.2. Chain Authentication Process

Algorithm 4 shows the process of chain authentication of the message by the receiver. For messages without a signature, the receiver needs first to determine whether the sequence number of the messages is legal and then authenticate the node value of the message hash chain of the messages. For messages with a signature, the receiver needs to authenticate the signature and verify the sequence number of the messages and the node value of the message hash chain. The parameters used in Algorithm 4 are described below:

- *chain_seq*: Sequence number counter.
- dk_{src}: Sender's public key.
- *HC_ver_node*: Message hash chain node value used for verification.

Algorithm 4 The Receiver Verifying The Received Messages

Input: Messages.
Output: Verification status .

Obtain header information, message sequence number, payload, and node value of the chain from the message: $hdr, seq, payload, HC\_pkt\_node \leftarrow DePkt(p)$.
if $chain\_seq + 1 \neq seq$ **then**
  return -1. # A value of "-1" indicates that the sequence number is not sequential.
end if
if $cur\_interval == Sig\_interval$ **then**
  $s \leftarrow Get\_Sig(p)$.
  if $Ver(dk_{src}, s) \neq TRUE$ **then**
    return -2. # A value of "-2" indicates an error in signature verification.
  end if
else
  $HC\_ver\_node \leftarrow HC\_Iteration(hdr, payload, HC\_lnode)$.
  if $HC\_ver\_node == HC\_pkt\_node$ **then**
    The sender inserts HC_ver_node at the end of the message hash chain and updates message hash chain tail node $HC\_lnode = HC\_ver\_node$.
    return 0.# A value of "0" indicates that the authentication of the message is successful.
  else
    return -3.# A value of "-3" indicates an error in message hash chain verification.
  end if
end if

7. Safety Analysis

The necessary definitions for proving the security of the message hash chain are given below.

Definition 1. *If there is always a μ_0 for all e such that $\varepsilon(\mu) < \frac{1}{\mu^e}$ when $\mu > \mu_0$, then $\varepsilon(\mu)$ is said to be a negligible value with μ as the parameter.*

Definition 2. *Note that H is the set of all hash functions, and h is a hash function. If h can find a, b, $a \neq b$, $h(a) = h(b)$ in polynomial time, then it is considered that h will have a hash collision. For $\forall h \in H$, if the probability of hash collision in h is equal to $\varepsilon(\mu)$, i.e., the probability of hash collision in h is negligible, then H is a non-collision hash function set.*

Definition 3. *Denote a digital signature scheme triple (Gen, Sig, Ver), which satisfies:*

- *Gen represents the asymmetric key generation algorithm. For the key pair (ek, dk), ek is the private key of the signature, and dk is the public key of the signature.*
- *Sig is the signature algorithm of the digital signature scheme. For the communication transmission sequence $p_1, p_2, p_3, \ldots$, there is $Sig(ek, HC_i, HC_i, HC_{i+1}, \ldots, HC_{i+q}) = s_i, s_{i+1}, \ldots, s_{i+q}$ on a certain segment of data transmitted, where q is a positive integer, and $HC_i, HC_{i+1}, \ldots, HC_{i+q}$ come from $p_i, p_{i+1}, \ldots, p_{i+q}$.*
- *Ver is the verification algorithm of the digital signature scheme. For the digital signature $s_i, s_{i+1}, \ldots, s_{i+q}$ of a certain segment of data and the (ek, dk) generated by Gen, there is always $Ver(dk, s_i, s_{i+1}, \ldots, s_{i+q}) = 1$.*

Definition 4. *For the digital signature scheme (Gen, Sig, Ver), if only the dk cannot forge the ek of the scheme in polynomial time, then the scheme is secure.*

Definition 5. *The message hash chain verification scheme (MHC, Gen, Sig, Ver) takes the digital signature scheme (Gen, Sig, Ver) as an option. On the basis of (Gen, Sig, Ver), it also satisfies:*

- *MHC is the construction algorithm of the message hash chain. For the transmission sequence* $p_i, p_{i+1}, \ldots, p_{i+q}$*, there is* $MHC(m_i, m_{i+1}, \ldots, m_{i+q}) = HC_i, HC_{i+1}, \ldots, HC_{i+q}$*, where* $m_i, m_{i+1}, \ldots, m_{i+q}$ *come from* $p_i, p_{i+1}, \ldots, p_{i+q}$*, respectively, .*
- *After receiving the sequence* $p_i, p_{i+1}, \ldots, p_{i+q}$ *and the* $HC_i, HC_{i+1}, \ldots, HC_{i+q}$ *encapsulated in it, the receiver also constructs a message hash chain node* $MHC(m_i, m_{i+1}, \ldots, m_{i+q}) = HC'_i, HC'_{i+1}, \ldots, HC'_{i+q}$ *for the received sequence through MHC, and there is* $\forall\delta$*,* $\delta \in (i, i+1, \ldots, i+q)$*,* $HC_\delta = HC'_\delta$*.*

Theorem 1. *The messages between two messages authenticated by chain signature also have authenticity and non-repudiation.*

(Gen, Sig, Ver) is a secure digital signature scheme, h is a known hash function, and the probability of hash collision at h is less than $\varepsilon(\mu)$, i.e., h is a non-collision hash function. In this case, if the digital signatures of p_α and p_β can be verified successfully and satisfy $\alpha < \alpha + 1 < \beta$, then $\forall\delta$, $\alpha < \delta < \beta$, p_δ can verify their authenticity and non-repudiation through massage hash chain verification.

Proof of Theorem 1. It is assumed that the message hash chain verification scheme (MHC, Gen, Sig, Ver) is insecure. This means that under the condition that (Gen, Sig, Ver) is a secure digital signature scheme and h is a non-collision hash function, the message hash chain verification cannot guarantee the authenticity and non-repudiation of the message sequence, which message sequence between the message p_α and the message p_β that can be successfully verified by (Gen, Sig, Ver). Then there is an attacker who uses algorithm $ATTCK$ to forge the (MHC, Gen, Sig, Ver) scheme, and obtains the signature sequence $S^{(1)}, S^{(2)}, \ldots, S^{(k)}$ transmitted by the victim and the message hash chain node value sequence $HC^{(1)}, HC^{(2)}, \ldots, HC^{(k)}$ according to the victim's dk, where $S^{(t)} = s_1^{(t)}, s_2^{(t)}, \ldots, s_m^{(t)}$, $HC^{(t)} = HC_1^{(t)}, HC_2^{(t)}, \ldots, HC_n^{(t)}$, $m, n \in N^*$ and $m < n$.

Then the scheme can output a valid signature sequence and message hash chain node sequence:

$$S^* = s_1^*, s_2^*, \ldots, s_r^*, \forall x \in \{1, 2, \ldots, j\},$$

$$S^* \notin S^{(x)} and S^* \neq S^{(x)},$$

$$HC^* = HC_1^*, HC_2^*, \ldots, HC_l^*, \forall y \in \{1, 2, \ldots, j\},$$

$$HC^* \notin HC^{(y)},$$

Specifically, algorithm $ATTCK$ uses Gen to generate a pair of (ek_{ATTCK}, dk_{ATTCK}), and then uses MHC to construct the message hash chain nodes $HC_1^*, HC_2^*, \ldots, HC_l^*$ of all message sequences $m_1, m_2, \ldots, m_l$. Finally, encapsulate them into the message sequence $p_1, p_2, \ldots, p_l$ of the message hash chain, and sign $p_1, p_2, \ldots, p_l$ with ek_{ATTCK}. The final output of algorithm $ATTCK$ is $S^* \notin S^{(x)}$ and $HC^* \notin HC^{(y)}$.

According to the assumptions, the signed and verified messages satisfy $Sig(ek, p_\zeta, p_\eta) = s_\zeta, s_\eta$ and $Ver(dk, s_\zeta, s_\eta) = 1, 1 < \zeta < \eta < r$. For $\forall\delta, \zeta < \delta < \eta$, p_δ only uses the message hash chain verification instead of the digital signature verification. Although the attacker cannot forge ek in $s_\zeta = Sig(ek, p_\zeta)$, it can forge its p_ζ as $p_\zeta^* = (m_\zeta, HC_\zeta^*)$. From $p_\zeta = (m_\zeta, HC_\zeta) = (m_\zeta, h(h(m_\zeta)||HC_{\zeta-1}))$, the following two situations will inevitably occur.

1. $HC_\delta^* = HC_\delta = HC'_\delta$. Obviously if $HC_\delta^* = h(h(m_\delta)||HC_{\delta-1}^*) = h(h(m_\delta)||HC'_{\delta-1})$, where h is a non-collision hash function, then $HC_{\delta-1}^* = HC'_{\delta-1}$ is obtained. Next, algorithm $ATTCK$ can output the message hash chain sequence

$$HC^* = HC_\delta^*, HC_{\delta+1}^*, \ldots, HC_\eta^*$$

and finally get $s_\eta^* = Sig(ek_{ATTCK}, HC_\eta^*)$. If there should be $Ver(dk, s_\eta) = 1$, but $Ver(dk, s_\eta^*) = 1$, it means that algorithm $ATTCK$ can forge (Gen, Sig, Ver) digital sig-

nature scheme. However, it obviously contradicts the assumption that (Gen, Sig, Ver) is a secure digital signature scheme.

2. $HC^*_\delta \neq HC_\delta = HC'_\delta$. Knowing that $HC^*_\delta = h(h(m_\delta)||HC^*_{\delta-1})$, there must be $HC^*_{\delta-1} \neq HC_{\delta-1} = HC'_{\delta-1}$ that can recursively get $HC^*_{\zeta+1} \neq HC_{\zeta+1} = HC'_{\zeta+1}$. For the message hash chains and digital signatures at both sides of the transmission corresponding to m_ζ, they satisfy the relational expressions $HC_\zeta = HC'_\zeta$ and $Ver(dk, s_\zeta) = 1$. If $HC^*_{\zeta+1} = h(h(m_{\zeta+1})||HC^*_\zeta) \neq HC_{\zeta+1}$, then the receiver can use the message hash chain to verify the authenticity and non-repudiation of $p_{\zeta+1}$, and then use the message hash chain to verify the authenticity and non-repudiation of p_δ in a recursive way, which contradicts the null hypothesis.

In summary, the null hypothesis does not hold. It means that under the condition that (Gen, Sig, Ver) is a secure digital signature scheme and h is a non-collision hash function, the message hash chain verification scheme (MHC, Gen, Sig, Ver) is secure. Therefore, the authenticity and non-repudiation of the data flow between two digital signature intervals can be ensured by using the message hash chain verification. □

Theorem 2. *Through the chain signature and authentication of a message, all messages in the previous sequence of this message can be verified*

Under the same conditions as Theorem 1, the receiver verifies the digital signature of a message p_α in the data stream. If $\forall\delta, 0 < \delta < \alpha$, then p_δ can judge its own authenticity and non-repudiation according to the correctness of p_α's digital signature.

Proof of Theorem 2. There is a sequence $p_1, p_2, \ldots, p_k$, the sender will sign the p_k, and the receiver will verify the signature. Suppose there is an attacker who can use algorithm $ATTCK$ to forge the node value of the message hash chain. This means that for the message hash chain sequences $HC = HC_1, HC_2, \ldots, HC_k$ and $HC' = HC'_1, HC'_2, \ldots, HC'_k$ constructed by $m_1, m_2, \ldots, m_k$, the algorithm $ATTCK$ can output the forged message hash chain node sequence $HC^* = HC^*_1, HC^*_2, \ldots, HC^*_k$ according to $m_1, m_2, \ldots, m_k$, and make $HC^* = HC$. In the absence of an attacker, when the receiver receives the $p_k = (m_k, HC_k, s_k)$, the verification of the signature must satisfy $Ver(dk, s_k) = 1$. If algorithm $ATTCK$ can output $s^*_k = Sig(ek_{ATTCK}, HC^*_k)$ to satisfy $Ver(dk, s^*_k) = 1$, then it means that algorithm $ATTCK$ can forge scheme (Gen, Sig, Ver), but this obviously contradicts the assumption. This shows that if p_k can be verified by digital signature, then $p_1, p_2, \ldots, p_{k-1}$ also has authenticity and non-repudiation; otherwise, $p_1, p_2, \ldots, p_{k-1}$ do not have authenticity and non-repudiation. □

Theorem 3. *The message hash chain can ensure the integrity and immutability of the data flow.*

Proof of Theorem 3. A message $m_i = (hdr_i, payload_i)$ and its corresponding node value of message hash chain HC_i are jointly encapsulated into a message hash chain message $p_i = (m_i, seq_i, HC_i)$, where hdr_i includes the source address src, destination address dst and other contents of the message header $other$. Obviously, the equation $HC_i = h(h(m_i)||HC_{i-1}) = h(h(hdr_i||payload_i)||HC_{i-1})$ can be obtained from the Formula (1). If any content of the message hash chain message is tampered with by an attacker, and the tampered values are hdr^*_i, $payload^*_i$ and HC^*_i, respectively, then $h(h(m^*_i)||HC'_{i-1}) \neq HC^*_i$ must occur when the receiver verifies it. □

8. Reliability Analysis

It is necessary to set the sequence number in the MHC method because the node values of the message hash chain should be calculated in strict order when constructing the chain. The difference between the sequence number contained in the message hash chain and that contained in IPSec is that the sequence number field is an optional field in IPSec, which is mainly used to provide anti replay services, while the sequence number field of MHC method is a necessary field, and each node of the message hash chain needs

to be constructed according to the sequence number. After IPSec establishes a SA for the first time or the SA reaches its life cycle to renegotiate parameters, it will clear the sequence number stored in the SA, and then incrementally count each message. The sequence number of the message hash chain inherits the previous changes and is not cleared, and the verification of each message must verify whether the sequence number changes incrementally in sequence. The reliability of message hash chain is mainly reflected in that the communication receiver should not only compare the sequence number to judge whether it is increased in order, but also verify the integrity, authenticity and non-repudiation of the whole message through the (MHC, Gen, Sig, Ver) scheme, and complete packet loss retransmission, chain synchronization and timely error detection through the sequence number.

8.1. Packet Loss Retransmission

If there is a data stream communication between the two communicating parties through the MHC method, the data stream $P = p_1, p_2, \ldots, p_n$ sent by A to B, each packet is $p_i = (m_i, seq_i, HC_i)$, where $m_i = (hdr_i, payload_i)$. The message hash chain constructed by the sender is HC, and the chain constructed by the receiver is HC'. Under the condition that the network has the possibility of packet loss, the following two situations must occur:

1. At least, there is a possibility that it is greater than $\varepsilon/2$, and the P received by B arrives in order, then the message hash chain $HC'_i = h(h(m_i||HC'_{i-1}))$ constructed by B through P satisfies $HC = HC'$.
2. At least, there is a possibility that it is greater than $\varepsilon/2$, and the data stream received by B may arrive out of sequence or lose packets. Assume that at a certain time t_0, the sequence number corresponding to the sender's tail node is seq_j, and the sequence number corresponding to the tail node used by the receiver for verification is seq_k, $k < j$. At this time, if the sender sends a new message p_δ to reach B, and its corresponding sequence number $seq_\delta > seq_k + 1$, then set the message retention time t_s for p_δ. Subsequently, at time t_1, where $t_0 < t_1 < t_0 + t_s$, if the message hash chain of the p_δ has not been successfully verified, the p_δ will be discarded, and the sender will request the following message corresponding to the sequence number of the current tail node of the chain. In contrast, if the chain of the p_δ can be successfully verified and the corresponding message hash chain is constructed at the receiver, the verification of the message corresponding to the last sequence number is continued.

8.2. Error Detection and Correction

The error detection function of the MHC method mainly uses the chain signature and chain synchronization mechanism to verify the message's integrity, authenticity, and non-repudiation in real-time by comparing the node values of the message hash chain in real-time and signing and authenticating the chain at intervals. If the attacker tampers or forges any message content, the verification of the node value and signature of the message chain will fail. Both communicating parties should re-request the message with verification error within a limited time to ensure that the data flow can achieve higher reliability or disable the illegal message sender to reduce network security risk.

9. Efficiency Analysis

9.1. Experimental Environment

The experiment uses C language to realize the MHC method of the network layer, and the experimental code runs on multiple PCs. The experiment uses the MHC method to set up the sender and receiver of network-layer data transmission. The PC configuration is Intel® Core™ i7-10875H CPU @ 2.30 GHz and 16 GB RAM. The TCP protocol of the transport layer does not contain additional settings, and its protocol header length is 20 B. The experiment compared the network layer's MHC method with the network layer's communication method using the traditional IP protocol, the AH protocol, and the ESP protocol of IPSec. The MHC method uses the SHA256 algorithm as the primary hash

function, while the IPSec uses the HMAC-SHA1-96 algorithm as the hash function used to calculate the HMAC. The asymmetric encryption algorithm in the (MHC, Gen, Sig, Ver) scheme is the RSA-2048 algorithm.

9.2. Efficiency Comparison of Several Communication Methods

In order to test the transmission efficiency of the MHC method, the experiment compared the efficiency of the network layer using the traditional IP protocol, the AH protocol, and the ESP protocol of IPSec with the method. At the same time, the communication method of IP protocol, which is signed and authenticated packet by packet to ensure data non-repudiation, is compared with the transmission efficiency of several other communication methods. The experiment set up five groups of test subjects. The transport layer of each group of experiments uses the TCP protocol. The network layer uses the IP protocol, the IP protocol with packet-by-packet signature and authentication, the AH protocol, the ESP protocol of IPSec, and the MHC method. We recorded the average number of messages transmitted by five groups of subjects in 2 min, 5 min, 10 min, 30 min, and 60 min, and the number of those in each group was the average of ten tests. The throughout capacity is then calculated based on the average amount of data transferred per group. Finally, the estimated throughout capacity is used as the standard to measure the transmission efficiency, and the efficiency of several groups of experimental objects is compared. The relevant information of the experiment is shown in Table 2.

$$Throughout\ capacity = \frac{Average\ data\ transmission}{Transmission\ time}.$$

Table 2. Information about test contents.

Serial Number	Transport Layer Protocol	Communication Mode	Payload Length of Each Message
1	TCP protocol	IP protocol	1460 B
2	TCP protocol	Packet by packet signed IP protocol	1372 B
3	TCP protocol	AH protocol	MTU-TCPH-IPH-AHH = 1436 B
4	TCP protocol	ESP protocol	MTU-TCPH-IPH-ESPH = 1436 B
5	TCP protocol	MHC method (signature interval 1000)	About 1420 B without signature and about 1332 B with signature

Among them, the AH protocol test group uses the transmission mode, and the identity authentication method uses certificate authentication. The life cycle of the first stage negotiation is 86,400 s, and the life cycle of the second stage negotiation is 120 s. AHH represents the AH protocol header, with a length of about 24 B; The ESP protocol test group also uses the transmission mode, and the identity authentication method uses certificate authentication. The life cycles of the first and second stages of negotiation are 86,400 s and 120 s, respectively. It is set to only verify the integrity of the message. ESP represents the ESP protocol header, with a length of about 24 B; The MHC method test group set its signature interval to 1000, and the other experimental settings are consistent with those in 6.2. The experimental test results are shown in Table 3.

The AH protocol experimental group uses the transmission mode, and the identity authentication method uses certificate authentication. The life cycle of the first-phase negotiation is 86,400 s, and the life cycle of the second-phase negotiation is 120 s. The ESP protocol experimental group maintains the same settings as the AH protocol experimental group, and at the same time, it only performs integrity verification on messages. The MHC method experimental group set the signature interval to 1000. MTU in Table 2 is the maximum transmission unit, and its length is 1500 B. TCPH is the TCP header, whose length is 20 B, while IPH is the IP header, the length is 20 B. MHCH is the MHC header, including the sequence number length of 4 B, the remaining fields of about 4 B, and the node value of the message hash chain length of 32 B, with a total length of 40 B. Finally, AHH is the AH protocol header with about 24 B, and ESPH is the ESP protocol header with about 24 B. The experimental results are shown in Table 3.

Table 3. Test results of transmission efficiency of five communication methods.

Network Layer Communication Method		2 min	5 min	10 min	30 min	60 min
IP protocol	Million packets	4.26	10.52	21.28	63.47	128.51
	Amount of data transmitted/Mb	49,809.46	122,837.52	248,540.61	741,347.66	1,501,003.01
	Throughout capacity/Mbps	415.08	409.46	414.23	411.86	416.95
IP Packet signature	Million packets	0.06	0.14	0.29	0.87	1.72
	Amount of data transmitted/Mb	641.96	1579.89	3217.72	9574.15	18,903.31
	Throughout capacity/Mbps	5.35	5.27	5.36	5.32	5.25
AH protocol	Million packets	1.25	3.12	6.2	18.42	36.98
	Amount of data transmitted/Mb	14,354.48	35,786.94	71,181.33	211,615.71	424,800.51
	Throughout capacity/Mbps	119.62	119.29	118.64	117.56	118
ESP protocol	Million packets	1.23	3.08	6.14	18.48	36.89
	Amount of data transmitted/Mb	14,151.8	35,376.74	70,483.29	212,260.47	423,755.83
	Throughout capacity/Mbps	117.93	117.92	117.47	117.92	117.71
MHC method	Million packets	1.31	3.3	6.58	19.79	39.45
	Amount of data transmitted/Mb	14,849.35	37,430.08	74,798.15	224,780.88	448,085.5
	Throughout capacity/Mbps	123.74	124.77	124.66	124.88	124.47

Figure 4 shows the comparison of the experimental results of the five groups of experiments. The results show that the transmission method whose network layer is IP protocol has the highest average transmission efficiency, and the average throughout capacity can reach more than about 400 Mbps. However, it has no additional security means and is prone to network attacks. Under the condition that only the transmission integrity is guaranteed, the average throughout capacity of the communication methods of the AH protocol and the ESP protocol is the same. The average throughout capacity of the MHC method is about 5% higher than that of the communication methods of the AH and the ESP protocols. The reason is that the MHC method directly uses the message hash chain for data authentication, and reduces the overhead of signature and verification through chain signature and authentication technology. However, the AH and ESP protocols require a two-stage key negotiation process before transmission. These two protocols renegotiate new parameters before the end of the life cycle of the second stage. At the same time, the negotiation process is expensive, and the processing speed of the messages is not significantly improved compared with the MHC method. After the negotiation is completed, these two communication methods have a slightly lower average throughout capacity than the MHC method. In unit time, the average throughout capacity of the MHC method is 4.96% higher than that of the AH protocol communication method and 5.70% higher than that of the ESP protocol communication method. Finally, under the condition that the transport layer is the TCP protocol, the average throughout capacity of the MHC method is about 23.5 times higher than that of the IP protocol and the packet-by-packet signature authentication method.

We also compared the transmission efficiency of the AH protocol communication method and the MHC method by recording the time it takes for both parties to transmit fixed-length data. We did not use the ESP protocol as a comparison object because the AH protocol has no encryption function and fewer irrelevant fields, making it easier to compare the transmission efficiency with the MHC method. Therefore, it is better to use the AH protocol as a comparison object instead of the ESP protocol. The experimental conditions were kept consistent with the above experimental conditions. In particular, the life cycle of the SA in the AH protocol's first stage is 86,400 s, and that of the SA in the second stage is set, respectively, at 2 min, 5 min, 10 min, and 60 min. In the experiment, the lengths of the data transmitted by the two communicating parties, respectively, were 1 G, 5 G, 10 G, and 50 G. The average value of five experiments is used to record the experimental results of each group. The experimental results are shown in Table 4.

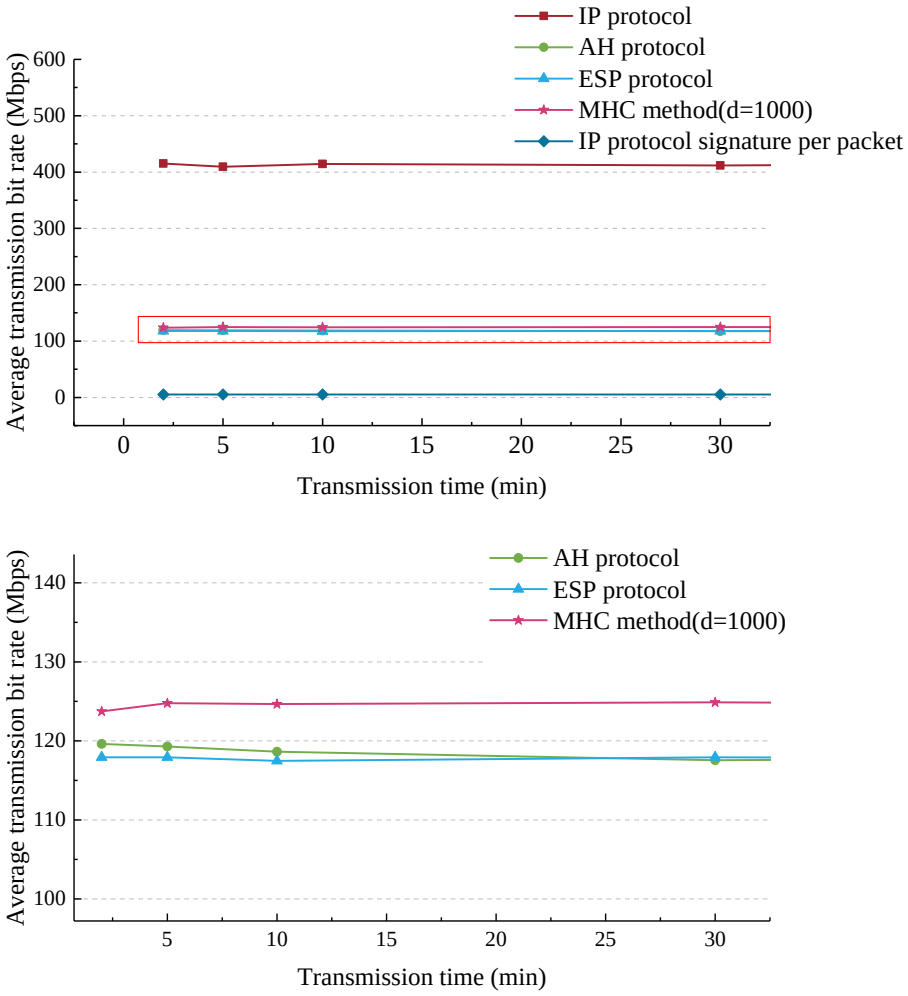

Figure 4. Comparison of average transmission efficiency of five communication methods.

The SA established by the AH protocol needs to set the life cycle and renegotiate and update the SA policy parameters before the end of the cycle. The shorter the life cycle, the higher the security of the parameters, but the negotiation process will affect the transmission efficiency. It can be seen from the test results shown in Table 4 that, under the above experimental conditions, the time used to transmit data of the same length in the MHC mode is shorter than that in the AH protocol communication mode. By calculating the average throughput of each test group, we found that the MHC method was more efficient than the AH protocol in each comparison group, and the larger the transmitted data, the more pronounced the gap. Specifically, taking the AH protocol communication method with a life cycle of 2 min as an example, when transmitting data with a length of 1G, the average throughput of the MHC method is 2.79% higher than the average throughput of the AH protocol, but when transmitting data with a length of 50G, this ratio increases to 5.77%.

9.3. Comparison and Analysis of Security Properties of Several Existing Schemes

In order to illustrate the security properties and efficiency of this scheme, this paper compares the chain network transmission mode using the MHC protocol at the network layer with the transmission mode using the IP protocol, IPSec protocol, and other existing schemes at the network layer. The differences in several security properties are shown in Table 5.

Table 4. The experimental results of the transmission efficiency of the AH protocol communication method and the MHC method.

Schemes and Configurations	Length of Transmitted Data			
	1 GB	**5 GB**	**10 GB**	**50 GB**
The MHC method/signature interval is 1000	68.76 s	337.47 s	678.31 s	3374.16 s
AH protocol/SA phase II life cycle is 2 min	70.68 s	358.53 s	712.34 s	3568.89 s
AH protocol/SA phase II life cycle is 5 min	70.75 s	353.60 s	704.51 s	3530.46 s
AH protocol/SA phase II life cycle is 10 min	70.03 s	351.51 s	701.97 s	3514.27 s
AH protocol/SA phase II life cycle is 60 min	70.00 s	351.58 s	698.37 s	3510.16 s

Table 5. Comparison of security properties of different schemes .

Scheme	Immutable	Integrity	Nonrepudiation	Reliability	Traceability	Synchronicity	Confidentiality	Efficiency
IP protocol	✗	✗	✗	✗	✗	✗	✗	Highest
IP protocol with signature [1]	✓	✓	✓	✗	✗	✗	✗	Lowest
AH protocol	✓	✓	✗	Higher	✗	✗	✗	Higher
ESP protocol	✓	✓	✗	Higher	✗	✗	✓	Higher
[11]	✗	✓	✓	✗	✗	✗	✗	Medium
[12]	✗	✓	✓	✗	✗	✗	✗	Higher
[13]	✓	✓	✓	✗	✗	✗	✗	Medium
[15]	✗	✓	✓	✗	✗	✗	✓	Lower
[16]	✓	✓	✓	Higher	✗	✓	✓	Low
MHC	✓	✓	✓	High	✓	✓	✓	Higher

[1] Sign and authenticate IP datagram packet by packet.

The IP protocol without a security mechanism has the highest transmission efficiency, but it does not have any security properties, which is easy to cause network attacks. After the IP protocol is signed and authenticated packet by packet, although the security of its transmission is improved, it also dramatically reduces the transmission efficiency. Both the AH protocol and ESP protocol of IPSec can ensure the integrity, non-tampering, and certain reliability of the messages, and the ESP protocol can also ensure the confidentiality of the messages. However, these two protocols cannot guarantee the non-repudiation of messages during the transmission process and are vulnerable to denial attacks by both parties. The rest of the schemes improve application-layer communication methods or use different hash chain structures and signature methods to improve security. Although the star-shaped and tree-shaped hash chain structure in the [11] can ensure that a signature can verify the child nodes under each tree, it cannot process packet loss data. The hash chain structure in the [12] improves the transmission efficiency, but it still cannot adapt to the network with the possibility of packet loss. The method in [13] caches the data, calculates its hash value, and then places the hash value in the message to be sent before verifying the later content with the previous content. This method needs to know the content of the entire transmission before transmission, which reduces the transmission efficiency and does not have security mechanisms such as reliability and confidentiality. The improved butterfly hash chain proposed in [15] has a complex structure, resulting in low transmission efficiency. The method in [16] is improved for the Modbus/TCP protocol at the application layer. It guarantees the confidentiality of the protocol through symmetric encryption and digital signature and can resist replay attacks by using a synchronization mechanism and a one-way guarantee scheme of a hash function. However, it still signs and authenticates each packet, resulting in low transmission efficiency. The MHC method ensures the integrity and immutability of the transmitted message through the hash chain. It uses the chain signature technology to realize the batch signature and authentication of the message stream, significantly reducing the overhead of signature and authentication. According to the characteristics that the message hash chain needs to be calculated, we designs the packet loss retransmission and chain synchronization mechanism to ensure the protocol's reliability and synchronization. It has the security properties of traceability and confidentiality.

10. Conclusions

The MHC method improves the traditional IP protocol. Using the improved MHC method to replace the traditional IP protocol can ensure that the network layer transmission has a security and reliability mechanism and the traceability of the message. The message hash chain can ensure the integrity, immutability, and synchronization of the transmitted data. At the same time, the use of chain signature and authentication technology can significantly reduce the overhead of signature authentication and improve the efficiency of secure transmission of message sequences. The MHC method has higher requirements on the reliability of the transmission process, so we design packet loss retransmission, error detection and correction, and chain synchronization for the communication process. Finally, the experimental results show that the MHC method adds an endogenous authentication mechanism and a reliable mechanism compared with the traditional transmission model without an authentication mechanism in the general software implementation. The MHC method can guarantee the non-repudiation of all previous messages through one signature. The transmission efficiency of the method is higher than that of the AH protocol and the ESP protocol of IPSec. Under the condition of ensuring the confidentiality of the transmitted message, the method has higher transmission efficiency than the ESP protocol. The method can make the transmission process more reliable and provide chain synchronization services for the transmission process.

In the future, we will further explore the impact of different hash functions and cryptographic algorithms on the efficiency and security of MHC methods. At the same time, we will also improve the network protocol stack based on the MHC method and try to implement the chip-level MHC method. Applying the MHC method to broadcast, Internet of Vehicles, aerospace, and other application scenarios is also the focus of our subsequent work.

Author Contributions: Conceptualization, M.H. and W.J.; methodology, W.J.; software, M.H.; validation, M.H. and W.J.; formal analysis, M.H.; investigation, W.J.; resources, W.J.; data curation, M.H.; writing—original draft preparation, M.H.; writing—review and editing, M.H.; visualization, M.H.; supervision, W.J.; project administration, W.J.; funding acquisition, W.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Program of China grant number 2018YFB1800100, and Scientific and Technological Innovation Service Capacity Building-Cyberspace Security Discipline Innovation Platform Construction Fund Project grant number 77F1910917.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. IP Security (IPsec) and Internet Key Exchange (IKE). Available online: https://www.rfc-editor.org/rfc/rfc6071 (accessed on 1 January 2020).
2. Dennis, F.; Martin, G.; Jörg, S.; Adam, C.; Marcin, S. The Dangers of Key Reuse: Practical Attacks on IPsec IKE. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018; pp. 567–583.
3. Kenneth, G.P. A Cryptographic Tour of the IPsec Standards. *Inf. Secur. Tech. Rep.* **2006**, *10*, 72–81.
4. Zhao, E.; Xiong, G. Reflective Denial-of-Service based on IKEv2 Protocol. *Commun. Technol.* **2019**, *52*, 144–148.
5. IP Authentication Header. Available online: https://www.rfc-editor.org/rfc/rfc4302 (accessed on 1 January 2020).
6. IP Encapsulating Security Payload (ESP). Available online: https://www.rfc-editor.org/rfc/rfc4303 (accessed on 1 January 2020).
7. Geetanjali, R.; Ashutosh, S.; Rajiv, K.; Farhan, A.; Razi, I. A trust management scheme to secure mobile information centric networks. *Comput. Commun.* **2020**, *151*, 66–75.
8. Adnan, M.; Quan, Z.S.; Sarah, A.S.; Subhash, S.; Wei, E.Z.; Hajime, S.; Wei, N. When Trust Meets the Internet of Vehicles: Opportunities, Challenges, and Future Prospects. In Proceedings of the IEEE 7th International Conference on Collaboration and Internet Computing, Atlanta, GA, USA, 13–15 December 2021; pp. 60–67.
9. Adnan, M.; Sarah, A.S.; Quan, Z.S.; Wei, E.Z.; Hajime, S.; Wei, N. Trust on wheels: Towards secure and resource efficient IoV networks. *Computing* **2022**, 1–22. [CrossRef]
10. Lamport, L. Password Authentication with Insecure Communication. *Commun. ACM* **1981**, *24*, 770–772. [CrossRef]

11. Chung, K.W.; Lam, S.S. Digital signatures for flows and multicasts. *IEEE/ACM Trans. Netw.* **1999**, *7*, 502–513. [CrossRef]
12. Golle, P.; Modadugu, N. Authenticating Streamed Data in the Presence of Random Packet Loss. In Proceedings of the NDSS Symposium, San Diego, CA, USA, 8–9 February 2001.
13. Liu, C. Research on Streaming Data Signature and Verification Based on Hash Chain. Ph.D. Thesis, Hunan University, Hunan, China, 2004.
14. Zhang, Z.; Sun, Q.; Wong, L. A proposal of butterfly-graph based stream authentication over lossy networks. In Proceedings of the 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlans, 6–8 July 2005; p. 4.
15. Miller, D. A Hash-Chain Based Method for Full or Partial Authentication of Communication in a Real-Time Wireless Environment. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2010.
16. Liu, F. Security authentication protocol of Modbus/TCP based on hash chain and synchronization mechanism. *Appl. Res. Comput.* **2018**, *35*, 1169–1173. 1186.
17. Huang, Q.; Huang, H.; Wang, W.; Li, Q.; Wu, Y. An Authentication Scheme Based on Novel Construction of Hash Chains for Smart Mobile Devices. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8888679. [CrossRef]
18. Zhang, H.; Zhu, Y. A Self-Updating Hash Chain Mechanism. *Wuhan Univ. (Nat. Sci. Ed.)* **2006**, *52*, 4–8.
19. Li, B.; Hou, Y.; Zhao, Y. An Optimized Scheme for Multicast Packet Authentication. *Comput. Eng.* **2006**, *32*, 3.
20. Hakeem, S.; El-Gawad, M.; Kim, H. Comparative Experiments of V2X Security Protocol Based on Hash Chain Cryptography. *Sensors* **2020**, *20*, 5719. [CrossRef] [PubMed]
21. Hakeem, S.; El-Kader, S.; Kim, H. A Key Management Protocol Based on the Hash Chain Key Generation for Securing LoRaWAN Networks. *Sensors* **2021**, *21*, 5838. [CrossRef] [PubMed]
22. Huang, N.; Zhu, J.; Guo, C.; Cheng, S.; Li, X. A Novel Hash Chain-Based Data Availability Monitoring Method for Off-site Disaster Recovery Architecture. *J. Circuits Syst. Comput.* **2021**, *6*, 2150294. [CrossRef]
23. Kim, D.; Seo, S.; Kim, H.; Lim, W.; Lee, Y. A Study on the Concept of Using Efficient Lightweight Hash Chain to Improve Authentication in VMF Military Standard. *Appl. Sci.* **2020**, *24*, 8999. [CrossRef]
24. Luo, G.; Shi, M.; Zhao, C.; Shi, Z. Hash-Chain-Based Cross-Regional Safety Authentication for Space-Air-Ground Integrated VANETs. *Appl. Sci.* **2020**, *12*, 4206. [CrossRef]
25. Han, M.; Jiang, W.; Guo, Y. Signature and authentication method based on message hash chain. *Appl. Res. Comput.* **2021**, *39*, 1183–1189.