Wuhan University Journal of Natural Sciences

2022, Vol.27 No.1, 001-010

Article ID 1007-1202(2022)01-0001-10

DOI https://doi.org/10.1051/wujns/2022271001

Hash-Based Signature for Flexibility Authentication of IoT Devices

□ **HAN Songshen[1], XU Kaiyong[1], ZHU Zhiqiang[1],GUO Songhui[1], LIU Haidong[1, 2], LI Zuohui[1]**

1. Graduate School, Information Engineering University, Zhengzhou 450001, Henan, China;

2. 722nd Research Institute, China Shipbuilding Group, Wuhan 430079, Hubei, China

Abstract: 5G provides a unified authentication architecture and access management for IoT (Internet of Things) devices. But existing authentication services cannot cover massive IoT devices with various computing capabilities. In addition, with the development of quantum computing, authentication schemes based on traditional digital signature technology may not be as secure as we expected. This paper studies the authentication mechanism from the user equipment to the external data network in 5G and proposed an authentication protocol prototype that conforms to the Third Generation Partnership Program (3GPP) standard. This prototype can accommodate various Hash-based signature technologies, applying their advantages in resource consumption to meet the authentication requirements of multiple types of IoT devices. The operation of the proposed authentication scheme is mainly based on the Hash function, which is more efficient than the traditional authentication scheme. It provides flexible and high-quality authentication services for IoT devices cluster in the 5G environment combining the advantages of Hash-based signature technology and 5G architecture.

Key words: Hash-based signature; secondary authentication; IoT device cluster; 5G

CLC number: TP 391

Received date: 2021-10-28

Biography: HAN Songshen, male, Master candidate, research direction: 5G authentication. E-mail: thordlos@foxmail.com

0 Introduction

In the IoT (Internet of Things) environment, various of IoT devices, from tiny, lightweight devices to powerful smart devices, are connected to send and receive information. The IoT can be divided into four layers: perception layer, transport layer, platform layer, and application layer. The transport layer has various connectivity technologies, such as WAN connectivity, including 4G/5G, eMTC, NB-IoT, Sigfox, LoRa, and LAN connectivity, including WiFi, Bluetooth, and ZigBee. The growing number of devices and the cooperation of multiple types of devices are important characteristics of the development of the IoT industry. The IoT under 5G scenario takes 5G technology as the core transport technology of the transmission layer of the IoT, makes full use of the technical advantages of 5G "high bandwidth, high capacity, high reliability and low delay", further transmits and exchanges the object information collected by the perception layer, realizing the interconnection between people and things, things and things. The 5G combined with SDN (Software-Defined Network), NFV (Network Functions Virtualization), virtualization, and other technologies, provides customized slicing services for IoT device clusters. According to the different needs of IoT devices clusters, the infrastructure provided by operators or vertical industry users can run these slices, reducing link costs and facilitating centralized control. Therefore, the IoT under the 5G scenario has the broadest application prospects. Figure 1 shows its representative application based on the three major application scenarios of 5G.

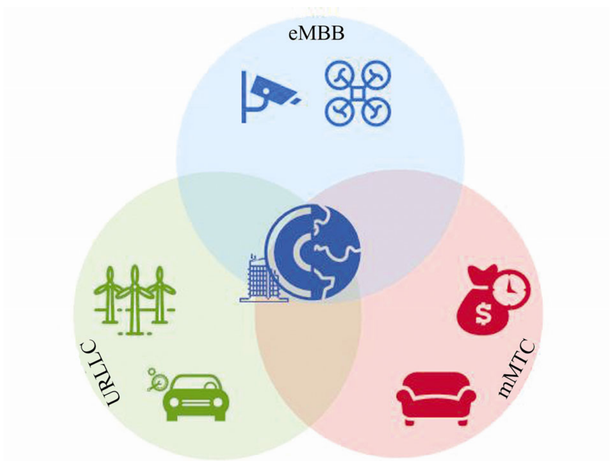


Fig. 1 IoT applications in the three major scenarios of 5G

Since IoT system contains many sensors and actuators, the security of IoT system is directly related to the security of information system, and even physical security. The main security threats of IoT systems under WAN include: data forgery, false data injection, replay attack, Sybil attack, etc. Relying on digital signature technology to achieve authentication between devices and ensure the integrity of communication data can cope with the above attacks.

5G provides a new security architecture with unified access authentication framework, security context and remote key management, and has stronger access device perception ability than traditional network environment. The service-based, virtualized and cloud-based 5G network architecture better supports the business needs of IoT devices, meanwhile, it has expanded the attack surface. Under the condition of network capacity opening, various devices are connected to the network. These devices can even manage and configure the 5G network, further exacerbating the risks caused by openness. Ensuring the legitimacy of the access device is the basis of security. Using digital signature technology to authenticate the equipment identity is more suitable for IoT devices. However, the traditional RSA, DSA (digital signature algorithm), and ECDSA (elliptic curves digital signature algorithm) signature algorithms have rigid computing requirements and hardly meet the authentication needs of IoT devices with different computing capabilities.

After years of development, Hash-based signature schemes now have relatively higher calculation speeds and relatively smaller signatures. The representative solutions have their characteristics and can be used as alternatives for IoT device authentication. The security parameters in the signature scheme can be adjusted, which increases the flexibility of authentication based on this scheme. On the whole, the Hash-based signature scheme has the following advantages:

1) General computing capabilities requirements. There are many different signature schemes (including stateful LMS (Leighton Micali Signature), XMSS (eXtended Merkle Signature Scheme), XMSS^{MT}; stateless FTS, SPHANCINS, SPHANCINS^{+}). Different signature mechanisms have different requirements for the computing capabilities of the device. The security parameters of each signature scheme can also be adjusted to accommodate various computing capabilities configurations of IoT devices cluster with different characteristics.

2) Minimal security assumptions[1]. Unlike other signature schemes, which rely on multiple complex operations to generate secure signatures, Hash-based solutions only require secure Hash functions. The security of the scheme is directly related to the Hash function used. If the Hash function used is proved to have a security risk, the security of the basic signature scheme will not be affected after the Hash function is replaced.

3) Quantum threat resistance. Unlike traditional cryptographic signature algorithms whose security depends on trap gate one-way functions based on the difficulty of decomposed integers and computed discrete logarithms, respectively, most Hash-based signatures are not susceptible to Shor's quantum algorithm[2]. This feature is more competitive under quantum threats, and it is also a reference indicator of whether a signature scheme is worth developing in the next few decades.

Therefore, Hash-based signature schemes are more competitive than traditional signature algorithms in the authentication of IoT devices. Our works include: Combining the research on 5G authentication schemes with the principle of Hash-based signature technology, analyzing the advantages of applying it to IoT device authentication in 5G scenarios, and then proposing an authentication protocol that applies Hash-based signature technology to terminal devices in the 5G scenario. This protocol can be integrated into a 5G unified identity management system to provide a template for applying Hash-based signature technology to authentication, which are more flexible for IoT devices with different computing capabilities.

1 Related Work

Some review papers have discussed in detail the application of Hash-based signature technology in the

IoT. They mainly spread out from the following aspects.

The trade-offs performance of Hash-based signature techniques during evolution: Butin[1] provided an overview of the post-quantum signature scheme and introduced the basic structure and evolution of the Hash-based signature scheme, and discussed the state and standardization of it. In Ref. [3], Palmieri analyzed the advantages of Hash-based signature schemes applied in IoT scenarios and the current difficulties, and then introduced the work progress of the cryptography community in the state and standardization of Hash-based signatures. Suhail *et al*[2] thoroughly discussed the key issues of migration to post-quantum signature schemes, analyzed the reasons for such transition, classifying Hash-based signature schemes into stateful and stateless according to key generation, signature process, and other construction parameters, and then discussed the advantages and disadvantages of each.

The feasibility and advantages of the Hash-based signature scheme applied to the IoT devices: Saldamli *et al*[4] evaluated the performance of Hash-based signature combined with Merkle's tree on Pyboard devices in terms of energy consumption, execution time, CPU usage, and memory consumption. They proposed that signing and validating could be handled on the Pyboard device with an acceptable processing time: 5 ms for signature and 1 ms for validation, but the memory required during key generation might not be satisfied. Given the limited processing performance of IoT devices, Pereira *et al*[5] designed a Hash-based signature scheme with smaller signatures based on the Winternitz signature scheme. Ghosh *et al*[6] designed a Hash-based signature method for resource-constrained IoT nodes based on XMSS and provided software and hardware collaborative implementation. Song *et al*[7] implemented key generation on FPGA (Field-Programmable Gate Array) to accelerate LMS, which supports all parameter sets of LMS. It provides technical reserve for the large-scale application of Hash-based signature in IoT devices.

In the other research field of application, Alzubi[8] proposed an authentication framework based on combining the Lamport-Merkle Digital Signature scheme and blockchain technology to solve the authentication problem in medical IoT devices. But the research only realizes "end-to-cloud" authentication in the traditional network environment. The Merkle signature-based variant used in this research has the same limitation as to the Merkle signature. The limitation means that public key can only sign a finite number of messages. Cho *et al*[9] combined the Hash-based signature mechanism with attribute-based access control to protect optical network devices in the SDN environment from malicious device security threats.

These researches attempt to apply the Hash-based signature technology to the authentication of IoT devices but only explore the application of one Hash-based signature technology without considering the advantages of the Hash-based signature system as a whole, which contains such a variety of technologies. The Hash-based signature scheme is more compatible with IoT devices than the traditional signature mechanism from resource consumption. The underlying Hash function between multiple signature schemes is universal, allowing multiple Hash-based signature schemes to exist in complementary forms in the same authentication process.

Also, these studies do not consider the impact that the network environment may have on the authentication itself. The high bandwidth of 5G can make the time delay caused by public keys and signatures with larger data volumes smaller in network transmission. In this case, the signature algorithm of identity authentication is mainly limited by the memory size of the IoT devices.

2 Authentication Mechanism for Accessing Data Networks

2.1 Secondary Authentication

After user equipment completes the identity authentication of the access network, the authentication for accessing a specific business service is secondary authentication. 5G supports optional secondary authentication between user devices and external data networks. 3GPP (Third Generation Partnership Program) standard TS 33.501[10] stipulates that the extensible authentication protocol (EAP) framework specified in RFC3748 should be applied to secondary authentication between the user equipment and the DN-AAA (Authentication, Authorization, and Accounting in external Data Network) server, as shown in Fig. 2.

Corresponding to the EAP framework, UE (User Equipment) is Supplicant, SMF (Session Management Function) is Authenticator, and DN-AAA is Authentication Server. In the interaction process of the secondary authentication protocol, network functions such as RAN (Radio Access Network), AMF (Access and Mobility Management Function), and UPF (User Plane Function) do not parse the specified authentication protocol, and these network functions only play a bearing role in the end-to-end authentication process. Before completing the secondary authentication, the UE has not really connected with the DN (external Data Network). The interaction between them is completely dependent on the forwarding of the bearer network. With this design separating the control plane from the data plane, malicious devices cannot access the external data network before completing identity verification, which cuts off the possibility of malicious traffic affecting the security of the original devices and services in the DN. Therefore, the integrity and compatibility design of identity authentication protocol are essential.

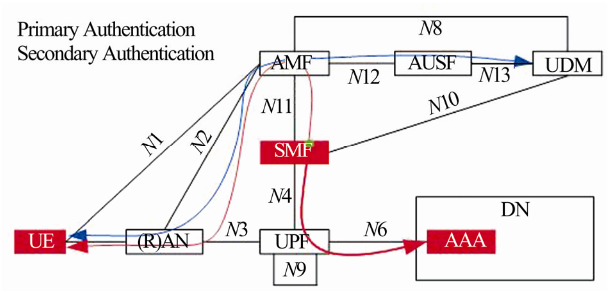


Fig. 2 Secondary authentication process in 5G

In the 5G SA (Standalone) stage, when the user has no special requirements for business security and trusts the operator, the existing authentication protocol can be selected to perform secondary authentication on the device. Users can also customize the authentication protocol and algorithms privately, and use security gateway authentication to achieve the purpose of security enhancement[11]. Authentication protocols are usually designed to assume that an attacker can have complete control over the traffic on the network and can participate in the protocol as a legitimate user. Attacks on authentication protocols are launched in many forms, and the two most commonly used are message replay attacks and MITM (Man-In-The-Middle) attacks. The design of the identity authentication protocol should focus on defending these two attacks.

When IoT devices with different computing capabilities are connected to the 5G network, the UPF establishes a connection for them to the DN only after the IoT devices have completed identity authentication. Using the traditional signature algorithm to authenticate the IoT devices, the IoT devices with limited computing capabilities may not be configured the secondary authentication, which will become a breakthrough for malicious devices to access DN and spread malicious traffic. Therefore, there is an urgent demand to provide devices with authentication adapted to their capabilities.

2.2 AKMA (Authentication and Key Management for Applications) Mechanism

5G introduces a new security mechanism for large-scale IoT deployment scenarios, AKMA[12], a cellular net-based delegated authentication system designated by the 3GPP. This new bootup architecture enhances existing solutions based on the GBA (Generic Boot Architecture) for 3G and 4G systems. 5G AKMA functionality does not need additional UE authentication. It reuses the results of 5G Primary Authentication to achieve mutual authentication between UE and DN. It means using the K_{ausf} established and stored in the Authentication Server Function (AUSF) and UE after successful 5G Primary Authentication to derive the shared key K_{akma} between the UE and AAnF (AKMA Anchor Function, a new network function for AKMK). This mechanism extends the trust relationship between universal subscriber identity module (USIM) cards and the operator network to the application layer. It avoids the security problem caused by key leakage when the initial key is generated and distributed on a large scale in the application layer.

Reusing K_{ausf} for AKMA eliminates the need to run AKMA-specific authentication separately from the Primary Authentication. The intermediate key of AKMA is calculated based on the existing key, which simplifies the AKMA program, helps reduce signaling, and saves on communication costs. It does not need to calculate from CK and IK (keys in EAP-AKA) like GBA, reducing the calculation cost of key derivation, which increases the range of IoT devices that AKMA functionality can serve.

5G terminals can safely establish or update the shared application layer key K_{AF} with different application servers/clouds through 5G AKMA function, derive new secondary service keys based on the key, and perform corresponding application layer security processing, establishing a secret key foundation for the security protection of new applications.

2.3 Comments

AKMA plays the same role as secondary authentication in the “UE-to-DN” authentication process. IoT devices use the derived symmetric key to encrypt communication with the external data network. Meanwhile, malicious users cannot communicate with the DN because they do not have this symmetric key. This result completes the factual authentication. However, AKMA depends on the specific implementation of the operator, and there may be an abuse of USIM cards. This practical problem brings management difficulties. On the other hand, a unified key deduction process does not provide differentiated services for IoT devices suited to their computing capabilities.

3 Hash-Based Signature Scheme

Different from the traditional signature scheme, this signature scheme mainly performs the Hash operation. Its security only depends on the security of the underlying Hash function. This feature is more prominent in one-time signature technology, which is the cornerstone of Hash-based signature technology. The Hash-based signature scheme can be instantiated with any Hash function meeting the security requirements, which are mainly focused on the following aspects: preimage resistance, second preimage resistance, and collision resistance. The underlying Hash function can be regarded as a block and replaced without changing the overall structure. In addition, when a specific Hash function is found to be vulnerable, it can be replaced directly. Hash operation and signature solution are not coupled. This feature eliminates the dependency on multiple security components, effectively reduces the complexity of implementation, and provides excellent flexibility to limited IoT devices, making it easy to deploy widely across IoT devices.

Because of no robust quantum algorithm to attack the Hash function, the Hash-based signatures are anti-quantum. The digital signature schemes commonly used in the existing designs of 5G mobile communication systems are RSA, DSA, and ECDSA. The security of these schemes relies on trapdoor one-way functions based on the hardness of factoring integers or computing discrete logarithms. With the practical application of Shor quantum algorithms, these hard problems in number theory are likely to be solved by quantum computers in polynomial time. However, most Hash-based signatures are not easily affected by the Shor algorithm. The most effective quantum attack on the Hash operation is the Grover algorithm, which provides progressive acceleration for brute force cracking of Hash function (including collision attack and preimage attack). But the impact on the security of the Hash function is much smaller than the impact of the Shor algorithm on prime factorization and discrete logarithms. The level of time consumption required to crack them varies between squares and cubes[13].

3.1 Lamport One-Time Signature

In a one-time signature (OTS) scheme, a private key pair can only be used to sign one message. First, the signer selects pairs of random numbers as private keys, in which each unit can be the number of bits output by the selected Hash function (i.e. 1 unit = 256 bits in SHA-256) or more. Then the signer calculates the Hash values of these random number units by the unit as the public key. To sign a message, the signer reads the message "bit-by-bit", and based on the value of that bit, selects a corresponding position unit from each private key pair. These selected units are ultimately combined into a signature of this message. The mechanism of the Lamport OTS scheme is shown in Fig. 3.

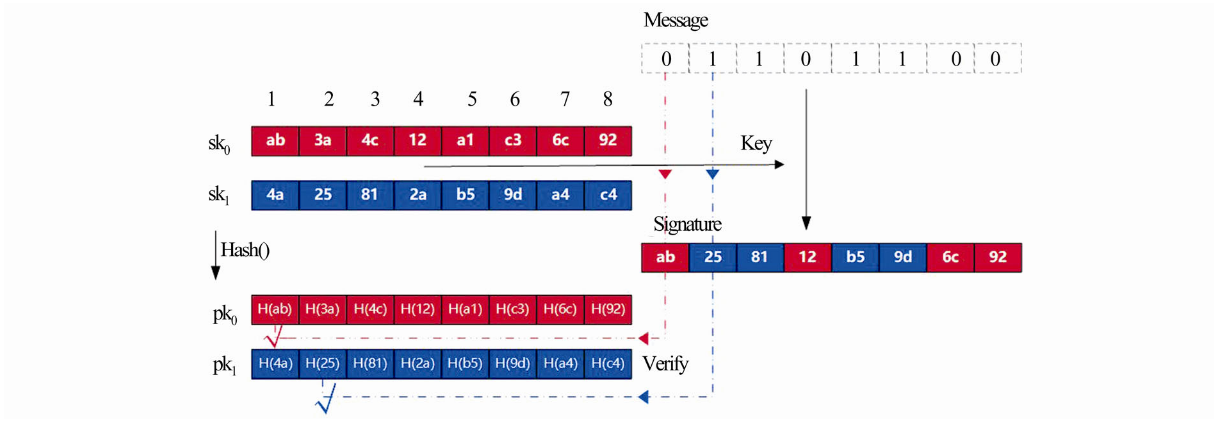


Fig. 3 One-time signature and signature verification

In this case, the signer needs to sign the message 01101100. If the "0-th" bit of the message is equal to 0, select the "0-th" unit from sk_0 sequence as the "0-th" unit of the signature. If the "1-th" bit of the message is equal to 1, select the "1-th" unit from sk_1 sequence as the "1-th" unit of the signature, and the rest can be done in the same manner. The verifier can then verify that if the Hash value of all signature units is equal to the value at the corresponding position in the public key, based on the original message. Once the "bit-by-unit" verification passes, it is proved that the person who signed the message is the private key holder. Note that the size of each color block in Fig. 3 is the number of bits of the Hash function output value chosen by the signature scheme, resulting in a huge signature obtained.

Obviously, the OTS key pair can only be used once. Because the signature actually exposes half of the private key pair, and a second signature using the same key exposes more information about the key pair, which gives an attacker the ability to tamper with the message bits with the private key unit of complementary bits. This prevents OTS from being used on a large scale in real-world scenarios.

3.2 Merkle Tree

The one-time signature was extended by the Merkle tree scheme. The central idea of this scheme is to combine a large number of one-time public key into a structure to obtain a value that can represent them. This value will be considered the public key of the Merkle tree. Based on a Hash tree structure, each message can be signed individually using a different OTS private key pair, up to 2^h messages can be signed, where h is the height of the Merkle tree.

This section refers to the example of the Merkle tree in Ref. [9] by Cho *et al*. The Merkle signature process is shown in Fig. 4.

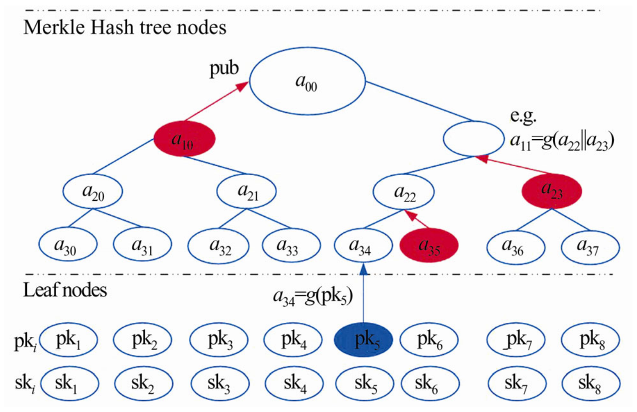


Fig. 4 Merkel tree mechanism

The Merkle tree needs to be fully generated before it can be used. The leaves are the public keys of one-time signature arranged in order. The lowest node of the tree is the Hash value of these leaf nodes, for example $a_{34}=g(\mathrm{pk}_5)$. Each internal node is the Hash value of its two child nodes joined together, expressed as $a_{xy}=g(a_{(x+1)(2y)}\,||\,a_{(x+1)(2y+1)})$, where $g()$ is a Hash function, || is connector. The root node pub at the top is the public key of the Merkle tree, which can be seen as a commitment, and leaf nodes may be revealed and proven to be part of this commitment.

When a given message M needs to be signed, the signer calculates its digest value $d=g(M)$ at first. Then, the signer selects the unused "i-th" one-time signature private key sk$_i$ to generate a signature σ_d for this message digest d, where $i\in(0,\cdots,2^h-1)$, h is the height of the tree, each OTS private key pair can only be used once. The signature result is sign= $(i,\sigma_d,\mathrm{pk}_i,\mathrm{Auth}_i)$, including the corresponding OTS public key pk$_i$, and the verification path Auth$_i$. The verification path Auth$_i$ consists of the complementary nodes of all nodes on the path from the "i-th" leaf node to the root. In the example in Fig. 4, $\mathrm{Auth}_5=\{a_{35},a_{23},a_{10}\}$.

Assume that the public key of the Merkle tree pub has been pre-distributed to the verifier. When the verifier receives the message M and the signature sign, it verifies the signature in two steps: Using the OTS public key pk$_i$ to verify the signature σ_d of the message digest d at first. If σ_d is a valid signature of d, calculate the root value by combining $g(\mathrm{pk}_i)$ and Auth$_i$. When the obtained root value matches the known public key pub, the signature sign should be accepted.

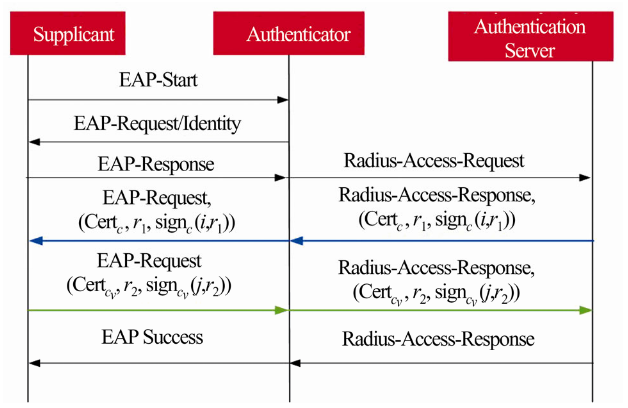
Without knowing the OTS private key sk, the attacker cannot forge the signature σ'_d for the false message digest d'. The Merkle tree uses one public key pub to provide legitimacy proof of the 2^h one-time signature public keys pk. The pk should be used to verify the correspondence between d and σ'_d as described in Section 3.1. It means that the first step cannot be verified successfully when an attacker does not have the correct OTS private key. When an attacker decides to forge one OTS key pair, since pk$'$ in sign$'$ is provided by the signer, the σ'_d will be misled into successful verification. However, since pk$'$ does not participate in the Merkle tree generation, the verification of the legitimacy of the OTS public key using pub and Auth$_i$ will be failed. It means that a forged OTS key pair will fail in the second step of verification.

3.3 The Proposed Authentication Protocol Prototype

Assume that there will be t IoT devices in a Data Network (DN) domain. IoT devices need to generate Merkle root public keys and use this to create a certificate. The certificate will be registered and distributed in a secure communication environment. After the public key pk$_v$ is derived from the IoT device v, the certificate issuer CA (certification authority) issues the following equipment certificate $\mathrm{cert}_v=(\mathrm{pk}_v,\mathrm{sign}_c)$, where $\mathrm{sign}_c=(i,\mathrm{OTS}_c(\mathrm{pk}_v),\ \mathrm{pk}_i,\mathrm{Auth}_i)$ and $i\in(0,\cdots,2^h-1)$. This certificate is distributed to the IoT device v along with the public key PK$_{\text{issuer}}$.

A Merkle tree with height h allows 2^h signatures to be signed using one Merkle root public key. Since DN domains typically require more signatures to be generated than IoT devices, there are $h_c \geqslant h_v$, and $0 < t < 2^{h_c}$. The Hash function $g()$ is determined according to the security level of the IoT device.

Assume that a mutual authentication connection needs to be established between the IoT device v and the external data network. The proposed authentication protocols are shown in Fig. 5.



Supplicant
Authenticator
Authentication Server
EAP-Start
EAP-Request/Identity
EAP-Response
Radius-Access-Request
EAP-Request, (Cert$_c$, r_1, sign$_c$(i,r_1))
Radius-Access-Response, (Cert$_c$, r_1, sign$_c$(i,r_1))
EAP-Request (Cert$_{cv}$, r_2, sign$_{cv}$(j,r_2))
Radius-Access-Response, (Cert$_{cv}$, r_2, sign$_{cv}$(j,r_2))
EAP Success
Radius-Access-Response

Fig. 5 Proposed authentication protocol prototype

They implement the following authentication protocols:

Step 1 Supplicant→Authenticator: EAP-start

The IoT device initiates the authentication by actively sending the request information to the Authenticator.

Step 2 Authenticator→Supplicant: Identity request

The Authenticator requests the identity of the IoT equipment.

Step 3 Supplicant→Authenticator→Authentication Server: IoT device's identity.

The IoT device responds with its own identity information. The Authenticator encapsulates it into a Radius-Access request message and transparently transmits it to the Authentication Server.

In this paper, the Hash-based signature algorithm is applied in the EAP framework. If there involves the negotiation of multiple Hash-based signature algorithms, the negotiation of the TLS protocol version in EAP-TLS can be referred. In addition, compared with the standardized EAP-TLS, this paper temporarily ignores the encryption algorithm negotiation, compression algorithm negotiation, etc.

Step 4 Authentication Server → Supplicant

AS (Authentication Server) generates a random number r_1, which is used only once. AS signs r_1 with a one-time signature private key pair, then sends $\{\text{cert}_c, r_1, \text{sign}_c(i, r_1)\}$ to the IoT device, where $\text{sign}_c(i, r_1) = (i, \text{OTS}_c(r_1), \text{pk}_i, \text{Auth}_i)$, and $i \in (0, \cdots, 2^{h_c} - 1)$.

When the signature verification is successful, the supplicant authenticates the identity of AS.

Step 5 After successful verification, IoT device v generates a random number r_2. v uses its OTS private key to sign r_2 and send $\{\text{cert}_v, r_2, \text{sign}_v(j, r_2)\}$ to AS, where $\text{sign}_v(j, r_2) = (j, \text{OTS}_v(r_2),\ \ \text{pk}_j, \text{Auth}_j)$, and $j \in (0, \cdots, 2^{h_v} - 1)$.

Note:

1) pk$_{\text{issuer}}$ has been pre-assigned to IoT devices;

2) As cert$_v$ corresponds to C, the cert$_c$ is issued by CA, and pk$_{\text{issuer}}$ can be used to verify the validity of the cert$_c$;

3) The public key of the one-time signature will be validated by the pk$_c$ in cert$_c$;

4) Only the person who holds the Lamport OTS private key can make the correct signature to r_1.

3.4 Security

The security of Hash-based signature schemes has been widely discussed, and it is obvious when the selected Hash function meets the security requirements.

Digital signature technology is widely used in authentication protocols to prove that the applicant is a private key holder in a way that does not expose private key information. The applicant sends the public key and uses the private key to sign his identity information to prove his identity, i.e.,

$$\text{sign}'(\text{Hash}(\text{ID}), \text{sk}),\ \text{pk} \tag{1}$$

where $\text{sign}'()$ is an asymmetric signature scheme, and $\text{Hash}()$ is secure Hash function.

However, this authentication scheme can only be used once if it were based on stateless signature technology because anyone intercepting communication messages can use the same information for authentication[14].

In a secure authentication protocol, not only the signature scheme should be tamper-proof and unforgeable, but also the design of the protocol should be resistant to replay attacks and man-in-the-middle attacks. This is also vulnerability in some researches using the Hash-based signature mechanism to complete authentication, such as Ref. [15]. Although they use the random number, token, or other mechanisms to improve the protocol's security, there are no research using a formal method to analyze their security.

Still some studies adopt traditional cryptographic methods, such as the Diffie-Hellman key exchange protocol, to support the authentication process in insecure channels. To some extent, that will lose the purity of Hash-based authentication, whose security is mainly guaranteed by Hash operation. Bellini *et al* avoids this problem by using a counter in Ref. [14], where the authentication client and server maintain a counting variable, respectively. The message of the client request authentication is

$$\mathrm{sign}'(\mathrm{Hash}(\mathrm{ID}\,||\,\mathrm{counter}),\mathrm{sk}),\ \mathrm{pk} \qquad (2)$$

where the counter is a variable starting from 0 and will increase after each authentication implementation success.

In this paper, the signature scheme adopted in authentication protocol is the primitive OTS combined with the Merkle tree. Since the Merkle tree is stateful, no additional counter is required.

The reason for calling it a prototype is that only by replacing the sign() (means $\mathrm{sign}_c()$, $\mathrm{sign}_v()$) function with a variant of other Hash-based signature schemes, the performance of the authentication scheme can be customized. The resource consumption of the authentication scheme depends on the selected signature algorithm. With multiple variants of Hash-based signature and their time-memory tradeoff characteristics, we apply it to the authentication of IoT devices cluster, which can fine-grainedly fit their computing and storage capacity.

4 Optimized Signature Technology

Combined with the Merkle tree method, multiple messages can be signed with a pre-distributed public key, which reduces the size of Lamport's one-time signed public key set and avoids the linear surge of public key cost. The disadvantage of this approach is that it can more than double the size of the signature. However, there are still some other mechanisms for optimizing the efficiency of signing and key usage.

4.1 Checksum Mechanism

In the Lamport method, by signing only bits with the value 1 in the message text, the entire sk_0 private key string can be completely discarded. That will be halving the size of the public and private keys. A checksum mechanism is introduced to avoid the security vulnerability that a signature can be forged by deleting several units in the signature, as shown in Fig. 6.

The checksum represents the number of "0-bit" in the original message. It is signed like the original message. When an attacker tries to modify the signature by removing a block, it equals an increase of the "0-bit" in

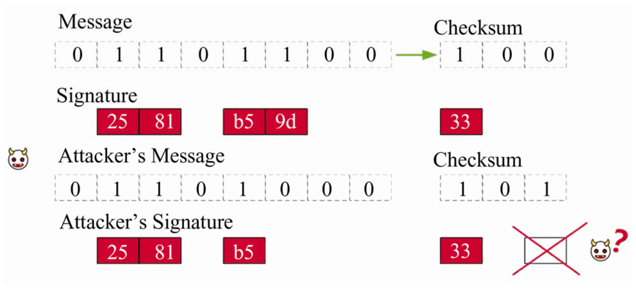


Fig. 6 Checksum mechanism

the original message, making the checksum invalid. The verifier will reject the modified signature.

The tamper-proof of the checksum will be guaranteed by the security of the signature mechanism itself. Suppose the attacker wants to increase the value of the checksum, it is necessary to convert the "0-bit" to the "1-bit" in checksum signature, which is impossible for the attacker who does not have the private key of the signature.

4.2 Winternitz Method

The core idea of the Winternitz one-time signature is to sign multiple bits of a message digest using only one unit of the private key. This mechanism can reduce the size of the signature to $1/n$ of the original, at the cost of an increase in the amount of calculation, where n is the Winternitz parameter.

Assuming that the one-time signature is signed directly on the byte, rather than bit-by-bit on the original message, this reduces the signature size by a factor of 8 (1 byte = 8 bit). The "unit-by-byte" signature does not map directly like the Lamport signature. Instead, the same key is iterated repeatedly according to the encoded value of the byte to generate the required Hash chain. Only one private key string needs to be stored, to control the number of times the Hash operation is executed on the same unit to implement the unit's signature.

Since the key columns are associated with each other, that is, $\mathrm{sk}_n = \mathrm{Hash}\ (\mathrm{sk}_{n-1})$, an attacker can tamper the signature in the direction of the increment of the byte block encoded value by doing the Hash function again. To resist this attack, the signer can calculate the checksum of the bytes of the original message and signs the checksum as well, preventing from signature tampering.

The messages that need to be signed in the authentication process are usually smaller. This method of increasing the amount of calculation to reduce the signature size is generally not applied separately to IoT devices.

4.3 Flexibility of Authentication Schemes

In combination with the mechanisms mentioned above, some powerful signature schemes have been proposed. XMSS[16] is one of more mature extension schemes for Merkle signatures. Compared with the original Merkle signature, XMSS generates smaller signatures at the same level of security. XMSS^{MT}, a variant of XMSS, can be used to sign an almost unlimited number of messages. Both LMS and XMSS use a variant of Merkle tree, and Winternitz-OTS. Campos *et al*[17] compared LMS and XMSS in detail on the same hardware platform Cortex-M4 and found that LMS has better performance. However, XMSS can use various acceleration schemes[17] to realize key generation and accelerate the process of signature and verification, thus it has better flexibility than LMS.

The research on Hash-based signature schemes is always a tradeoff between signature size, key size, and running time. As shown in Fig. 7, the signature scheme that is most suitable for specific IoT device capabilities can be selected as the basis for identity authentication in these variants, providing adaptive authentication schemes for IoT device clusters and integrating these authentication schemes into a unified authentication framework.

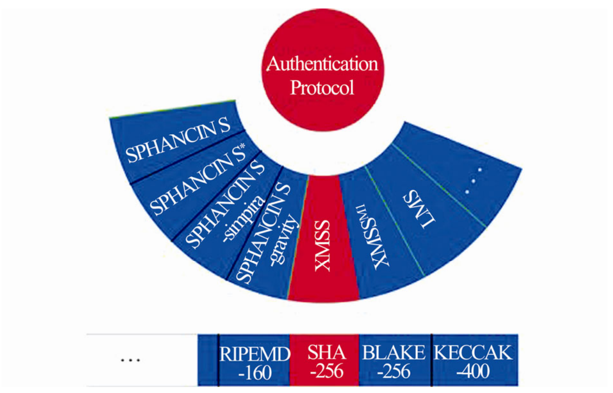


Fig. 7 Flexibility of authentication protocols

From top to bottom, the signature protocol in the authentication protocol can be selected flexibly. Under certain safety strength requirements, SPHANCINS, XMSS, LMS, etc. can be selected appropriately. The selection is based on computation and memory requirements, and the parameters of the selected signature mechanism can also be adjusted. Signature algorithms between different trust domains can be diverse, but the determination of the one inter-trust domain authentication protocol needs to consider the universality.

Moreover, deploying different Hash functions at different security levels in one trust domain, can reduce the potential security threat posed by a single function. XMSS supports SHA-256, SHA-512 as Hash functions. Hash functions of unstandardized signature algorithms can be customized by users, which increases the potential for Hash-based signature to be used in complex authentication systems.

4.4 Efficiency Comparison with Traditional Protocols

Compared with the traditional RSA and ECDSA signature scheme, Hash-based signature can provide more fine-grained adaptation to IoT devices with different capabilities, avoiding weak authentication or cancellation of authentication due to insufficient computing capabilities of IoT devices, and can alleviate the identity security risk of the IoT cluster under the same slice, which is caused by the authentication shortboard.

In Ref. [18], the authors implement a signature protocol based on the classic OTS and Merkle trees. Through theoretical analysis and experiments on mobile devices, it is found that Hash-based signature improves the efficiency of keys generating, the signing and verification processes.

In recent years, research on Hash-based signature technology itself like Ref.[19] has focused on the use of pseudo- random generators, and the reduce of key size. The use of multi-trees increases the number of signatures and reduces the time of generation of signature and verification keys. For IoT devices with severely limited capabilities, FPGA can be used additionally to provide computing capabilities support like Ref. [6].

5 Conclusion

This paper proposes a Hash function-based signature scheme for the authentication of IoT devices in 5G environment and analyzes the effect achieved by Hash-based signature, which has the potential to make full use of the advantages of the Hash function-based signature scheme.

At present, it can only determine the variant as the basis of identity authentication according to whether the capability of the IoT device meets the existing signature scheme. The signature implementation process of some typical variants can be further analyzed to determine which computations can be offloaded on edge computing nodes in a 5G environment, providing better support for devices with limited capabilities. If the Hash operation is implemented in an infrastructure way to provide services for the upper-layer signature and authentication applications, the resource consumption of the terminal device can be further reduced.

References

[1] Butin D. Hash-based signatures: State of play [J]. *IEEE Security and Privacy*, 2017, **15**(4): 37-43.

[2] Suhail S, Hussain R, Khan A, *et al*. On the role of Hash-based signatures in quantum-safe internet of things: Current solutions and future directions [J]. *IEEE Internet of Things Journal*, 2020, **8**(1): 1-17.

[3] Palmieri P. Hash-based signatures for the Internet of Things: Position paper [C]//*Proceedings of the* 15*th ACM International Conference on Computing Frontiers*. New York: ACM, 2018: 332-335.

[4] Saldamli G, Ertaul L, Kodirangaiah B. Post-quantum cryptography on IoT: Merkle's tree authentication [C]// *Proceedings of the International Conference on Wireless Networks* (*ICWN*). Berlin: Springer-Verlag, 2018: 35-41.

[5] Pereira G C C F, Puodzius C, Barreto P S L M. Shorter Hash-based signatures [J]. *Journal of Systems and Software*, 2016, **116**: 95-100.

[6] Ghosh S, Misoczki R, Sastry M R. Lightweight post-quantum-secure digital signature approach for IoT motes [EB/OL]. [2021-11-10]. *https*://*eprint.iacr. org*/2019/122.*pdf*.

[7] Song Y, Hu X, Wang W, *et al*. High-speed and scalable FPGA implementation of the key generation for the leighton-Micali signature protocol [C]//2021 *IEEE International Symposium on Circuits and Systems* (*ISCAS*). Washington D C: IEEE, 2021: 1-5.

[8] Alzubi J A. Blockchain-based Lamport Merkle digital signature: Authentication tool in IoT healthcare [J]. *Computer Communications*, 2021, **170**: 200-208.

[9] Cho J Y , Szyrkowiec T. Practical authentication and access control for software-defined networking over optical networks [C]// *Proceedings of the* 2018 *Workshop on Security in Softwarized Networks*: *Prospects and Challenges*. New York: ACM, 2018: 8-13.

[10] 3GPP. 3GPP TS 33.501 Version 15.4.0 Release 15 Security Architecture and Procedures for 5G System[S/OL]. [2019-12-20]. *https*://*www.etsi.org*/*deliver*/*etsi_ts*/133500_133599/133501/ 15.04.00_60/*ts*_133501*v*150400*p.pdf*.

[11] Wang J Y, Lv S L, Xu J M. Analysis of secondary authentication methods for vertical industries that can be applied to 5G Networks [J]. *Communications Technology*, 2020, **10**: 2538-2542.

[12] 3GPP. 3GPP TS 33.535 Version 16.0.0 Release 16 Authentication and Key Management for Applica-tions(AKMA) Based on 3GPP Credentials in the 5G System (5GS) [S/OL]. [2020-02-25]. *https*://*www.etsi.org*/*deliver*/*etsi_ts*/133500_133599/ 133535/16.00.00_60/*ts*_133535*v*160000*p.pdf*.

[13] Bernstein D J. Cost analysis of Hash collisions: Will quantum computers make sharcs obsolete [J]. *SHARCS*09, 2009, **9**: 105-109.

[14] Bellini E, Caullery F, Hasikos A, *et al*. You shall not pass! (once again) an IoT application of post-quantum stateful signature schemes [C]// *Proceedings of the* 5*th ACM on ASIA Public-Key Cryptography Workshop*. New York: ACM , 2018: 19-24.

[15] Singh N, Chhabra G, Singh K P, *et al*. A secure authentication scheme in multi-operator domain (SAMD) for wireless mesh network [C]//*Proceedings of the International Conference on Data Engineering and Communication Technology*. Berlin: Springer-Verlag, 2017: 343-357.

[16] Buchmann J, Dahmen E, Hulsing A. XMSS——A practical forward secure signature scheme based on minimal security assumptions [C]//*International Workshop on Post-Quantum Cryptography*. Berlin: Springer-Verlag, 2011: 117-129.

[17] Campos F, Kohlstadt T, Reith S, *et al*. LMS vs XMSS: Comparison of stateful Hash-based signature schemes on arm cortex-m4 [C]//*International Conference on Cryptology in Africa*. Berlin: Springer-Verlag, 2020: 258-277.

[18] Lizama-Pe´rez L A, Montiel-Arrieta L J, Herna´ndez- Mendoza F S, *et al*. Public Hash signature for mobile network devices [J]. *Ingenieria Investigaciony Tecndogia*, 2019(2): 1-10. DOI: 10.2220//FI.25940732E.2019.20n2018.

[19] de Oliveira A K D S, Lopez J, Cabral R, *et al*. High performance of Hash-based signature schemes [J]. *International Journal of Advanced Computer Science and Applications*, 2017, **8**(3): 421-432.

□