electronics

MDPI

Article

A Business Process-Based Security Enhancement Scheme for the Network Function Service Access Procedure in the 5G Core Network

Ying Zhu [1,*], Caixia Liu [1,2], Wei You [1], Yiming Zhang [1] and Weicheng Zhang [1]

[1] National Digital Switching System Engineering & Technological R&D Center, PLA Strategic Support Army Information Engineering University, Zhengzhou 450001, China
[2] Institute of System Engineering, Academy of Military Sciences, Beijing 100089, China
* Correspondence: 15068941803ying@alumni.sjtu.edu.cn; Tel.: +86-18930078586

Abstract: As the signaling processing center of 5G, the security and stability of the 5G Core Network (5GC) are of great importance for 5G. The current 5GC consists of multiple mutually independent Network Functions (NFs). However, the NF service access procedure does not match NF service requests and business processes. NFs can request authorized services for access at any time, which poses a security threat to NFs and user data. This paper proposes a security enhancement scheme for NF service access procedures based on the business process, which realizes the management of the NF business process. The NRF adds a token identifier field bound to the business process in the access token and establishes an access token repository to store the token identifier. NF Service Producer introduces an access token re-signature mechanism and a shared repository of responded access tokens. The security of the proposed scheme is verified by theoretical analysis and formal analysis, and the performance of the proposed scheme is evaluated in terms of response rate and resource consumption. The experimental results show that the proposed scheme can meet the security requirement with little efficiency degradation under the condition of increasing certain resource loss.

Keywords: 5GC; Network Function; access token; access control; AVISPA

check for updates

Citation: Zhu, Y.; Liu, C.; You, W.; Zhang, Y.; Zhang, W. A Business Process-Based Security Enhancement Scheme for the Network Function Service Access Procedure in the 5G Core Network. *Electronics* **2023**, *12*, 576. https://doi.org/10.3390/electronics12030576

Academic Editor: Nuno Cruz

Received: 22 December 2022
Revised: 18 January 2023
Accepted: 20 January 2023
Published: 23 January 2023

1. Introduction

Compared with the 4th Generation Mobile Communication Network (4G), the 5th Generation Mobile Communication Network (5G) has significantly improved the connection density, processing rate, and service quality. 5G realizes the separation of the user plane and the control plane and the design of the decentralized core network architecture. The Service-Based Architecture (SBA) is an essential feature of the 5G core network (5GC), whose core is to disassemble the network elements in 5GC into mutually independent and specific Network Functions (NFs). During the deployment of 5GC, NF can be freely encoded to meet the diversified requirements of different application scenarios. 5G networks are compatible with legacy 2G, 3G, and 4G connections and provide an interface for User Equipment (UE) to connect to the Internet. The complex network connection structure poses a new challenge to 5G security, and 5G security is also related to the security and stability of the whole network structure.

The security threats of 5GC include the Distributed Denial of Service (DDoS) attacks, man-in-the-middle attacks, and eavesdropping [1–4], which will pose a significant threat to the reliability of 5GC and the privacy protection of messages. As the signaling processing center of 5G, the safety and reliability of 5GC are very essential for the regular operation of 5G. As a basic unit of 5GC, NFs use the Service-Based Interface (SBI) for communication. NFs collaborate to realize the business functions of 5GC. The business functions of 5GC can be specifically broken down into a set of associated NF service accesses. Based on the NF service access procedure defined by the 3rd Generation Partnership Project (3GPP), NF can request access to other NF services, so ensuring the security of the NF service access procedure is the basis to ensure the regular operation of 5GC.

There are some prior works analyzing the security threats in the 5GC. Ahmad et al. proposed that 5GC is the main target of security threats and is prone to security vulnerabilities [2]. Cao et al. analyzed the security of 5G SBA and proposed that attackers may eavesdrop, tamper with communication messages between NF, or launch man-in-the-middle attacks [4]. Bhuiyan et al. tested the security of SBI and found that some interfaces have security threats, such as hijacking attacks, authorization bypass, and malicious redirection [5]. The literature [6] analyzed the possible security risks associated with the open-enabled interfaces of the 5GC. It proposed the Internet web technology adopted in 5G SBI architecture has many security vulnerabilities and the security of Application Programming Interface (API) functions, such as the Network Exposure Function (NEF), shared by 5GC to the outside may become a problem. Zhang et al. analyzed the architecture of 5GC and pointed out that the current 5GC faces DDoS attacks and privacy disclosure risks [7]. Zhang et al. studied the NF service access procedure defined by the 5G network protocol. They proposed that attackers can invoke the NF service illegally if the access token is leaked [8].

The protocol advancement group and scholars have also designed many security mechanisms to deal with the security issues in 5G. The 3GPP protocol advancement group has made a comprehensive consideration of 5G security when developing the 5G standardization scheme. They analyzed the trust model of the current 5G, the hierarchy of keys, and the access authentication mechanism of the current 5G [9]. Fang et al. summarized the challenges and future directions of 5G wireless security and proposed a new 5G wireless security architecture. They analyzed the identity management and flexible authentication of the proposed architecture [10]. Zhang et al. summarized the security mechanisms adopted in the current 5G SBA, including UE authentication mechanisms, secure SBIs, and secure network exposure interfaces provided by NFs [7]. Ferrag et al. provided a comprehensive review of network authentication and privacy protection schemes for 5G cellular networks. They summarize these schemes from three aspects: encryption method, human factor, and intrusion detection method [11]. Zhang et al. designed an access token enhancement scheme for the service access procedure to resist replay attacks on the token by introducing random and sequence factors into the access token [8].

Automated protocol validation tools automatically detect the presence of security vulnerabilities in protocols. Scholars have used formal analysis tools to analyze security vulnerabilities in protocols and verify the effectiveness of the proposed schemes. Arapinis et al. modeled and analyzed the security of 3G protocols using a formal approach to expose new threats to user privacy in 3G, proposed a remediation scheme for this problem and verified the effectiveness of the remediation scheme by a formal analysis tool [12]. Hussain et al. used the ProVerif tool to analyze security threats in LTE networks [13]. Basin et al. used Tamarin Prover, a security protocol verification tool, to conduct a comprehensive and systematic security assessment of 5G security models [14]. Hu et al. analyzed the current 5G forensic authentication protocol using Tamarin Prover and found the flaws in the current protocol and fixed them [15]. Automated Validation of Internet Security Protocols and Applications (AVISPA) is an automated protocol analysis tool that simulates the attacker's attack path with the stolen information. Dhillon et al. verified the security of the lightweight hash and heterogeneous operations proposed in the paper using the formal analysis tool AVISPA [16]. Chen et al. proposed a cryptography-based security protocol for edge computing scenarios and verified the security of the proposed scheme by theoretical analysis and AVISPA [17]. Zhang et al. proposed a fast-switching authentication scheme for the 5G mobile edge computing scenario and verified the security of the proposed scheme using AVISPA [18].

The service access procedure of NF is the basis of the regular communication of NF, and there is relatively little analysis of the security of NF service access. In this paper, we analyze the NF service access process. We find that the Network Repository Function

(NRF), as the authorization center, verifies that the NF-requested service is authorized, but does not verify that the NF-requested service matches the process of the 5GC business function. For example, an Access and Mobility Management Function (AMF) can invoke a Unified Data Management (UDM) service that is authorized to be accessed at any time, regardless of whether the accessed User Equipment (UE) is in its service area. In addition, the access token issued by the NRF allows the NF to repeatedly initiate service access requests to the NF Service Producer within a valid time. When the token is leaked or the communication message is eavesdropped on, the attacker can launch a replay attack on the producer within the validity of the token. This brings security risks to NF Service Producers, as well as the risk of leakage of UE privacy data stored in NF.

To solve this problem, this paper proposes a business process-based access security enhancement scheme for NF services with the access token as the core. Specifically, it includes: (1) adding a token identifier field bound to the business process in the access token, introducing a static business process chart and a dynamic access token repository to achieve systematic management of the token identifier; (2) adding a business information field and a UE identity field in the NF request message to achieve fine-grained management of NF access privileges; (3) designing a Producer re-signed token mechanism to ensure that token information can be effectively passed to the following process; (4) limiting the possibility of repeated access to the Producer by the NF through a shared access token repository. Finally, the security of the proposed scheme is verified using the automatic protocol verification tool AVISPA, and the performance of the proposed scheme is tested experimentally.

The contributions of this paper are as follows:

1. We analyze the security risks in the NF service access procedure in 5GC. Firstly, the NRF does not verify whether the business process matches the service request. Secondly, repeated access with the access token held by NF will bring replay attack risks to the NF Service Producer.
2. We propose a security enhancement scheme for NF service access procedure based on business processes and design a series of security mechanisms, from the token request message to the Producer's response to the service request, to achieve the management of NF business process and access rights.
3. We verify the safety of the proposed scheme by theoretical analysis and formal analysis tools and demonstrate the scheme's performance by experimental simulation.

2. 5GC NF Service Access Procedure Security Analysis

2.1. Procedure Overview

NF service access procedure involves three subjects: NF Service Consumer, later referred to as the Consumer, NRF, and NF Service Producer, later referred to as the Producer, and NF service access is achieved by invoking the Producer's service operation or accessing the Producer's service resources. The specific NF service access procedure is shown in Figure 1.

Step 1: The Consumer requests an access token from the NRF. The request message contains the type of the expected Producer and the requested NF service, with the identity information of the Consumer itself.

Step 2: When the NRF receives the request, it first verifies whether the Consumer has access to the target NF service. If the Consumer is authorized to access the requested service, the NRF generates an access token with appropriate claims included. The token states the Producer's identity information and the service authorized to be accessed by the Consumer, and the NRF uses the private key to sign the token.

Step 3: If the authorization is successful, the NRF sends the access token to the Consumer.

Step 4: The Consumer initiates a service access request to the Producer with an access token.

Step 5: The Producer verifies the integrity of the token using the public key of the NRF. Then the Producer verifies the claims in the token, including the validity claim, scope claim, and audience claim.

Step 6: If the verification of the Consumer is successful, the Producer opens the corresponding permission to the Consumer according to the request message. The Producer shall execute the requested service and respond to the Consumer's service request.

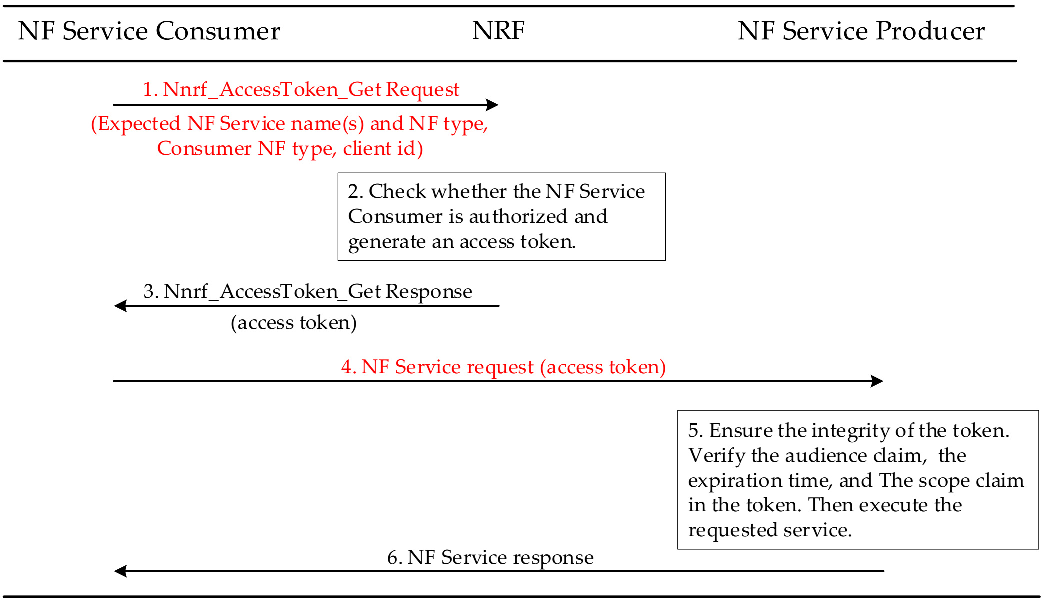

NF Service Consumer
NRF
NF Service Producer
1. Nnrf_AccessToken_Get Request
(Expected NF Service name(s) and NF type, Consumer NF type, client id)
2. Check whether the NF Service Consumer is authorized and generate an access token.
3. Nnrf_AccessToken_Get Response
(access token)
4. NF Service request (access token)
5. Ensure the integrity of the token. Verify the audience claim, the expiration time, and The scope claim in the token. Then execute the requested service.
6. NF Service response

Figure 1. NF service access procedure [19].

The access token authorizes access to the Consumer at the basic range of service levels. Depending on the configuration of the Producer, the granularity of authorization that the NRF opens to the Consumer in the access token can be refined to a specific service operation or a specifically requested resource. At the same time, security checks are added by the Producer, and the Producer can perform a secondary authorization of the Consumer's identity and the service operation requested by the Consumer.

2.2. Procedure Security Analysis

Although the 3GPP protocol refines the authorization scope when defining the NF service access procedure, they restrict the services or operations that NF can access while holding the token [19,20]. However, the NRF lacks the business logic rationality verification of NF service requests when it authorizes service access, and such malicious behaviors as NF unauthorized access are still operable. Consumers can access a range of services beyond what they need to authorize access by requesting multiple tokens for accessing other services. Specific security issues are analyzed as follows:

1. The NRF authorization does not verify that the service requested by the Consumer matches the business process. The request message defined by the protocol lacks UE identity information and business information. The Consumer can request service authorization from the NRF outside the business procedure at any time. The security risks caused by eavesdropping and malicious access between NFs are not fully considered.
2. The access token issued by the NRF declares the token validity time field. The Consumer can repeatedly initiate access to a single or a group of Producers that match the audience claim during the token validity time. The Producer's resources may be accessed illegally, and services may be invoked maliciously. If an attacker steals the access token or hijacks a legitimate NF, such as through eavesdropping, the Producer and the NRF may experience a replay attack.

The communication pattern of "Request-Response" between NFs follows the flow of request first and response second. The authentication procedure of the UE is used as an example to describe the business procedure and business logic rationality of the service request. The specific operation processes of the authentication procedure of UE are shown in Figure 2. The abbreviations used in Figure 2 are listed in Table 1. In a complete authentication procedure, the triggering sequence of the NF service is fixed. The Authentication Server Function (AUSF) sends the UE authentication request to the UDM only after it receives the UE authentication request from the AMF.

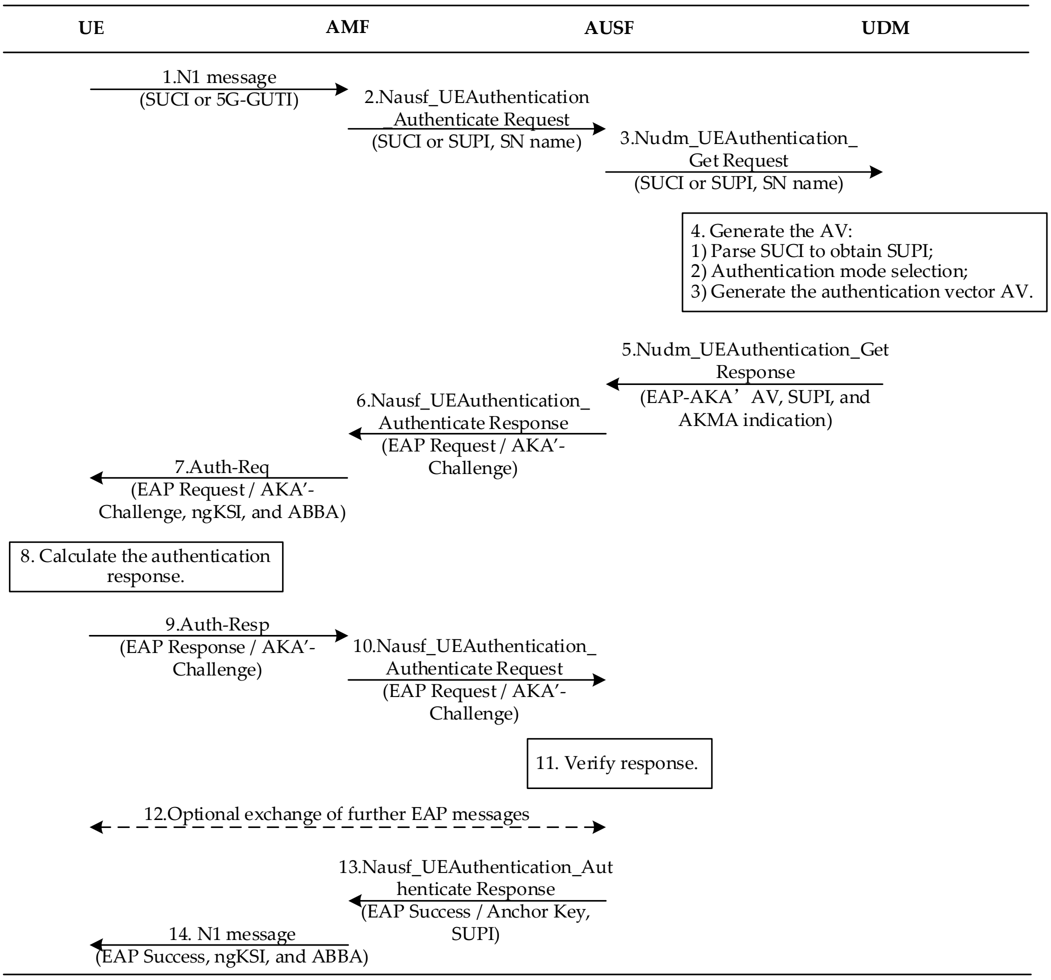

UE
AMF
AUSF
UDM
1.N1 message
(SUCI or 5G-GUTI)
2.Nausf_UEAuthentication _Authenticate Request
(SUCI or SUPI, SN name)
3.Nudm_UEAuthentication_ Get Request
(SUCI or SUPI, SN name)
4. Generate the AV:
1) Parse SUCI to obtain SUPI;
2) Authentication mode selection;
3) Generate the authentication vector AV.
5.Nudm_UEAuthentication_Get Response
(EAP-AKA' AV, SUPI, and AKMA indication)
6.Nausf_UEAuthentication_ Authenticate Response
(EAP Request / AKA'- Challenge)
7.Auth-Req
(EAP Request / AKA'- Challenge, ngKSI, and ABBA)
8. Calculate the authentication response.
9.Auth-Resp
(EAP Response / AKA'- Challenge)
10.Nausf_UEAuthentication_ Authenticate Request
(EAP Request / AKA'- Challenge)
11. Verify response.
12.Optional exchange of further EAP messages
13.Nausf_UEAuthentication_Aut henticate Response
(EAP Success / Anchor Key, SUPI)
14. N1 message
(EAP Success, ngKSI, and ABBA)

Figure 2. UE Authentication procedure [21].

Table 1. Some Abbreviations in 5G.

Abbreviation	Description	Abbreviation	Description
SUCI	Subscription Concealed Identifier	5G-GUTI	5G-Globally Unique Temporary Identifier
SUPI	Subscription Permanent Identifier	SN name	Service Network (SN) Name
EAP	Extensible Authentication Protocol	AKA	Authentication and Key Agreement
AV	Authentication Vector	AKMA	Authentication and Key Management for Applications
ngKSI	Key Set Identifier in 5G	ABBA	Anti-Bidding down Between Architectures

However, in the current UE authentication procedure, the AUSF can still request the UE authentication service from the UDM without receiving a Nausf_UEAuthentication_Authenticate Request message from the AMF. Because the AUSF is authorized to access the UDM's authentication service, the AUSF can both repeatedly request the access token from the NRF and repeatedly initiate service access to the UDM while the token is valid. If the AUSF repeatedly sends a wrong SUPI, SN name information in the request message for accessing the UDM will consume a lot of computational resources of the UDM. At the same time, it may also cause the UDM to fail to generate authentication vectors and make errors. Such request messages that do not conform to the business process can threaten the security and reliability of the UDM and pose a risk of leakage of the UE privacy data stored in the UDM.

The NRF only verifies that the service requested by the Consumer is authorized but does not verify that the current service request of the Consumer is consistent with the business process. Each NF can initiate service access repeatedly at any time, which creates conditions for replaying attacks, resulting in threats to the security of 5GC, and creating conditions for malicious NF to steal UE information, leading to the disclosure of sensitive UE information. Therefore, the primary research purpose of this paper is to propose a safe and reliable security enhancement scheme for service access procedures. First, the NRF can verify whether the service request of NF is consistent with the business process to avoid the repeated authorization of the NRF to the Consumer. Second, restrict the access rights of the Consumer to the Producer after obtaining the token to ensure that NF cannot request services beyond the business process from the Producer and cannot request specific services from the Producer repeatedly when accessing services.

In the deployed 5GC, an NF multithreading processes request messages from different NFs and initiate different service access requests. An NF participates in multiple service access processes, and an NF acts as both the Consumer and the Producer simultaneously. The NRF has an important authorization role in the service access procedure, and NFs must first request an access token from the NRF when requesting services from other NFs. The current service access procedure implements the control of the Consumer's access rights through the NRF to declare accessible services in the access token or specify the service operation information that can be invoked.

The access token is the key to delivering service authorization information, so this paper takes the access token as the basis of the security enhancement scheme design. The starting point of the scheme is to introduce new mechanisms around the service access authorization function of the NRF to strengthen further the NRF's verification of the token request message. To achieve the target of the NRF to verify the business logic rationality of the request message, we adjust the principles involved in the procedure. The Consumer adds the business information field and the UE identity information field to the request message. The NRF identifies and distinguishes the access token based on the business information. Meanwhile, the NRF passes the business information and authorization information to the Producer by adding corresponding claims to the access token.

In the specifically designed verification scheme, the NRF must implement the management of the NF business process, so that it can verify the business logic rationality of the Consumer. First, the NRF adds a unique token identifier bound to the business information into the token claims when generating an access token. The NRF establishes an access token repository to store the issued access tokens. Upon receiving a request from the Consumer to obtain an access token, the NRF additionally verifies that its requested service conforms to the business process. The Producer re-signature token is also designed to pass the current process business information to the following process to prevent Consumers from forging business processes. In addition, the access token repository shared by the Producer is designed in this scheme. The Producer stores the token identifier of the responded token in the shared token repository, which can effectively counteract the privacy leakage risk caused by repeated access by the Consumer.

3. Improved NF Service Access Procedure

3.1. Overall Framework

In this paper, the current NF service access procedure is improved so that the NRF can verify not only the legitimacy of the access request but also the business logic rationality of the access request. This section describes in detail the implementation of service request business logic rationality verification.

The improved interaction relationship between the NFs is shown in Figure 3, and the basic procedure is as follows.

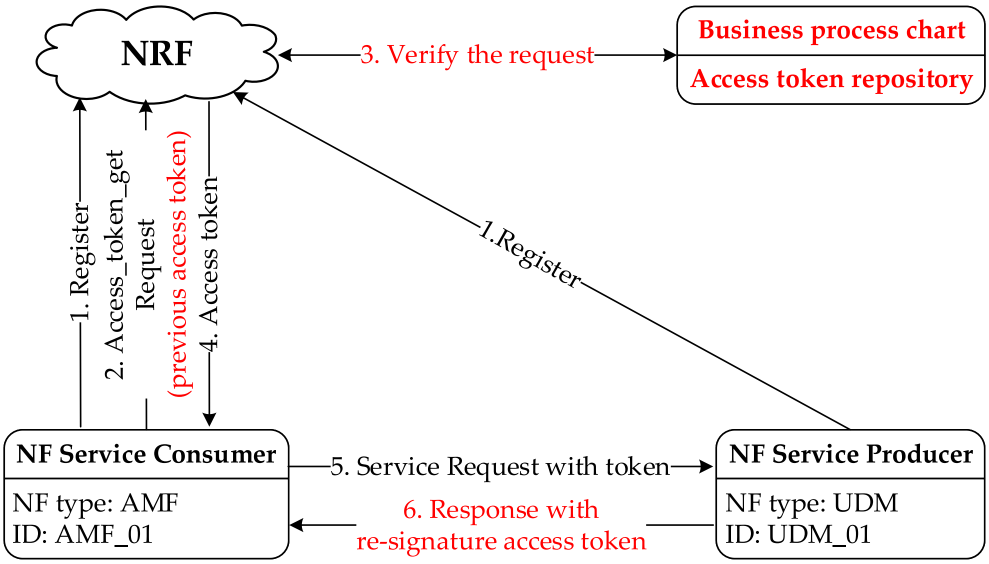


Figure 3. NF Interaction Relationship.

Step 1: NF initiates a registration request to the NRF when it becomes operative for the first time. The NRF establishes a static business process chart and a dynamic access token repository.

Step 2: The Consumer requests an access token from the NRF. The Consumer adds the UE identity field, the corresponding business information field, the Producer identity in the previous process field, and the access token used in the previous business process to the request message.

Step 3: The NRF checks whether the Consumer is authorized to access the requested service. The NRF checks the request message and the access token in the request message. First, verify the integrity of the token. Second, verify that the information in the token matches the information in the request message. Third, verify that the token is the latest token for the current business. Fourth, verify that the service invoked by the following process corresponding to the previous token is the same as the service requested in the request message.

Step 4: After verification, the NRF responds to the Consumer's request for an access token, generates the corresponding access token, and issues it to the Consumer. The token identifier field, UE identity information field, and business information field are added to the token claims, where the token identifier of the access token is bound to the business process.

Step 5: The Consumer carries the access token to request access from the Producer. The Consumer adds the UE identity information field and business information field to the service request message.

Step 6: When responding to the Consumer request, the Producer uses its private key to re-sign the token and returns the re-signed token to the Consumer.

The relevant symbols are described in Table 2, and the functions are described in Table 3.

3.2. Consumer Requests NRF Authorization

The Consumer generates a request message GAT to the NRF for the token.

$\mathrm{GAT} = \mathrm{Kn}(\mathrm{CONS}||\mathrm{PROS}||\mathrm{5GCPP}||\mathrm{UID}||\mathrm{REQS}||\mathrm{RAT}')$,
$\mathrm{RAT}' = \mathrm{PAT}||(\mathrm{PAT})\_\ \mathrm{Kps}'$,
$\mathrm{PAT.CLA} = \mathrm{CONS}'||\mathrm{PROS}'||\mathrm{5GCPP}'||\mathrm{UID}'||\mathrm{REQS}'||\mathrm{Ta}'||\mathrm{Te}'||\mathrm{JTI}'$,
$\mathrm{PAT.RIA} = (\mathrm{CONS}'||\mathrm{PROS}'||\mathrm{5GCPP}'||\mathrm{UID}'||\mathrm{REQS}'||\mathrm{Ta}'||\mathrm{Te}'||\mathrm{JTI}')\ \mathrm{Kns}$,
$\mathrm{JTI}' = \mathrm{5GCPP}'||\mathrm{PID}'||\mathrm{EID}'$,
$\mathrm{PAT} = \mathrm{PAT.CLA}||\mathrm{PAT.RLA}$.

Table 2. Symbol Description.

Symbol	Description	Symbol	Description
NFA	NF Service Consumer	NFB	NF Service Producer
5GCPP	Four decimal digits indicate 5GC business procedure, including UE registration procedure, PDU session establishment procedure, and so on.	UID	UE identity information such as SUPI, SUCI, PEI, GUTI, etc.
CONS	Consumer identity information, including NF type, instance ID, etc.	PROS	Consumer identity information, including NF type, instance ID, etc.
REQS	The Producer service requested by the Consumer	GAT	The message from the Consumer requesting an access token from the NRF
RSQM	The message from a Consumer requesting services from a Producer	RESP	Response of the NRF/the Producer to the Consumer request message
CLA	The claims in the access token	RIA	The signature in the access token
AT	The access token (CLA||RIA)	PAT	The access token of the previous process
Ta	The generation time of the access token	Te	The expiration time of the access token
PID	Business number of a specific business	EID	Business process number
JTI	Token identifier (5GCPP||PID||EID)	RAT	The re-signature token by the Producer
Kc	The public key of the Consumer	RT	The time when a Consumer requests a service from a Producer
Kn	The public key of the NRF	Kns	The private key of the NRF
Kp	The public key of the Producer	Kps	The private key of the Producer
K(A)	Use key K to encrypt content A	D_K(M)	Use key K to decrypt ciphertext M
(A)Ks	Use the private key Ks to sign content A	RV_K	Use key K to verify the RSA signature
AUTH	The service requested by the Consumer is authorized	NAUTH	The service requested by the Consumer is not authorized
NIL	The address indicated by the token identifier does not store the token	NFID	The own identity of the Producer
.	Subfield	||	Symbol of connection
′	Symbol of the previous process	∈	The dependency $A \in B$ means that A is a subset of B.
=	Equal sign	if	Judgment symbol

Table 3. Function description.

Name	Description
F1	The NRF verifies that the Consumer is authorized to access based on the Producer's profile. Input: CONS||PROS; Output: AUTH or NAUTH.
F2	The NRF queries the service request, Producer, and Consumer of the next business process from the static business process chart. The input is divided into two cases: (1) Input: (5GCPP||CONS||PROS||REQS) of the previous process; Output: (CONS||PROS||REQS) of the new process. (2) Input: 5GCPP; Output: (CONS||PROS||REQS) of the first process of the business.
F3	The NRF queries the current process of a business from the access token repository. Input: 5GCPP||PID; Output: EID.
F4	The Producer verifies that tokens are stored in the address space in the shared token repository. Input: JWT; Output: NIL or NFID.
EXE	The Producer performs the service requested by the Consumer. Input: REQS, Output: RESP.
GET	The NRF gets the business number of a particular business from the dynamic token repository. Input: 5GCPP, Output: PID.
MODA	The NRF updates the token identifier and business process stored in the access token repository. In 5GCPP||PID corresponding storage space to store new EID. Input: 5GCPP||New PID||EID.
MODB	The Producers update the contents of the Shared token repository, storing the (JTI)Kps||NFID in the corresponding storage space of JTI. Input: JTI.
MODC	The NRF updates the business number PID of the specific business in the token allocation table for the access token repository and stores the token identifier in the storage space indicated by the token identifier. Input: 5GCPP||New PID||EID.

Any 5GC business can be subdivided into multiple interrelated NF service processes with a fixed trigger order. The Consumer, the Producer, business, UE, and service request information of the previous process is passed into PAT and RAT′. The corresponding information of the current process is passed into GAT.

The improvement scheme proposed in this paper refines the authentication granularity of a request message to the specific service request while carrying the access token used by the previous process. With the addition of UID, 5GCPP, and RAT fields in the token request message, the NRF can verify whether the requested service matches the business process. At the same time, the NRF can pass more information to the Producer through the access token so that the Producer can perform secondary authorization for the access request of the Consumer.

3.3. The NRF Verifies the Token Request Message of the Consumer

3.3.1. Business Process Chart

Bulleted lists look like this:

Taking the UE authentication procedure as an example, the business processes that require authorization by the NRF are extracted from the protocol procedure represented in Figure 2, and a static UE authentication process chart is created at the NRF, as shown in Table 4.

Table 4. UE authentication process chart.

Process	Consumer-Producer	Requested Service	Operation
Step 1	AMF to AUSF	Nausf_UEAuthentication_Authenticate	POST
Step 2	AUSF to UDM	Nudm_UEAuthentication_Get	GET
Step 3	AMF to AUSF	Nausf_UEAuthentication_Authenticate	PUT

The NRF constructs the business process chart based on the business procedure defined in the 3GPP protocol [21]. Upon receiving a message from a Consumer requesting an access token, the NRF can match the request message with the process information in the business process chart based on the business information and the access token of the previous process in the request message.

From the business information and business process number indicated by the token identifier in the access token, the NRF can locate the Consumer of the current business process. Then the NRF can check the request message as follows:

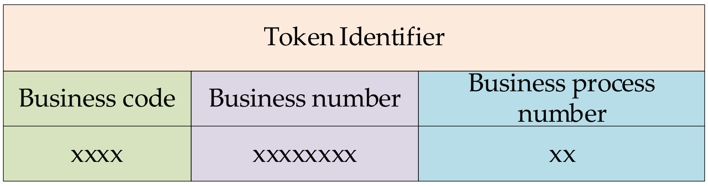
(1) Whether the Consumer identity type and the Producer identity type in the request message are consistent with the corresponding information specified in the process chart.
(2) Whether the service requested in the request message is consistent with the service corresponding to that process defined in the process chart.
(3) Whether the service operation requested in the request message is consistent with the corresponding service operation in the process chart.

After checking the business process based on the business process chart, the NRF queries the dynamic access token repository to see if the latest process of the business is the business process indicated by the access token.

3.3.2. Access Token Repository

As shown in Figure 4, the token identifier consists of three parts: business code, business number, and business process number. The business code is represented by 5GCPP, which is used to distinguish different services in 5GC, and the service name of each service in 5GC corresponds to the service code, respectively.

The business number is represented by the PID, which is used to distinguish the actual execution of the same business in 5GC. In any period, 5GC needs to process different business requests from UEs. In the 5GC business initiation phase, the NRF numbers the business where the service request is located based on the specific content of the Consumer request. After the initialization of the access token repository, the business number is an eight-digit decimal number, sequentially numbered from 00000001, and the business numbers are not shared among different 5GC businesses.



Token Identifier		
Business code	Business number	Business process number
xxxx	xxxxxxxx	xx

Figure 4. Token identifier.

The business process number is determined by the number of times the business requests an access token from the NRF, and the business processes are numbered sequentially starting from 01. When 5GC business is executed, NF service access occurs sequentially. Then the business process number information in the token identifier is the same as the process information in the static process chart.

The business ID consists of a business code and a business number together to distinguish specific businesses executed in 5GC and to ensure that each actual executed business has a unique number.

Business ID = Business code | | Business number.

Based on the above design, the token identifier can identify the actual process of each specific business. To speed up the finding speed of the token identifier, this paper designs the access token database into two parts: the token allocation table and the hash table. The latest business number of 5GC business is stored in the token allocation table, and the latest business process number with the complete token identifier is stored in the hash table.

A Hash Table is a data structure that enables direct access to a Key-Value by mapping it to an address in a table, as shown in Figure 5. Any element in the Key Value space X in the Hash Table M has a uniquely determined counterpart in the address space Y.

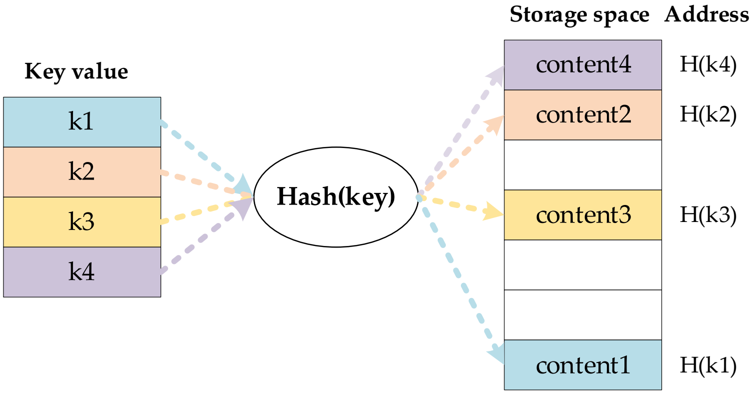
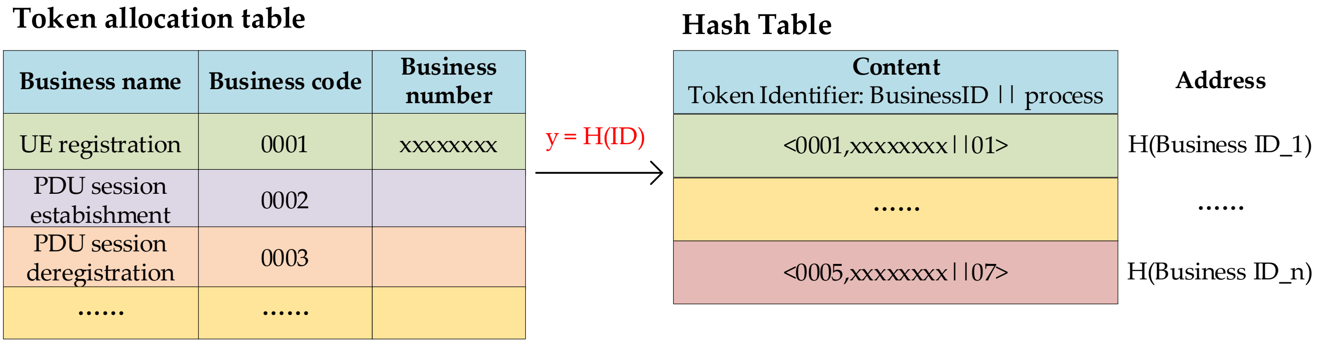


Figure 5. Schematic of hash function mapping.

The token allocation table is stored as a sequential table. The token allocation table stores the business codes and the current latest business numbers of the businesses. The business number is known to be sequentially numbered, and the NRF adds one to the business number corresponding to the business code in the token allocation table when processing the subsequent request initiated by the business.

Alfoudi et al. introduced distributed hash tables to manage many mobile devices to maximize network performance [22]. The introduction of hash tables enables fast lookup of token identifiers. The *business ID* is used as the keyword for hash calculation, and the obtained hash value y is the storage address of the token, as shown in Figure 6. The storage space, indicated by the storage address y, stores the complete token identifier, including three fields: business code, business number, and business process number.



Token allocation table

Business name	Business code	Business number
UE registration	0001	xxxxxxxx
PDU session estabishment	0002	
PDU session deregistration	0003	
……	……	

Hash Table

Content Token Identifier: BusinessID | | process	Address
<0001,xxxxxxxx||01>	H(Business ID_1)
……	……
<0005,xxxxxxxx||07>	H(Business ID_n)

Figure 6. Storage of Token Identifiers in the NRF.

When the NRF receives an access request from an NF carrying a previous token requesting a token, the authentication procedure for the Consumer request message is as follows.

Step 1: The NRF verifies the integrity of the previous process access token.

Step 2: The NRF checks the business process based on the business process chart and checks whether the service requested by the Consumer is authorized.

Step 3: Based on the token identifier of the previous process access token, the NRF quickly locates the business process executed by the Consumer and the business ID of that business.

Step 4: The NRF finds the business number corresponding to the business code in the token allocation table. Because the business number of the token is sequentially numbered, if the business number in the table is greater than or equal to the business number in the token, that is, the business ID of the token is accurate and reliable.

Step 5: The NRF hashes the business ID to get the storage address of the business process. If the business process number stored in the hash table is the same as the business process number indicated by the token identifier, the NRF passes the previous process access token verification.

Step 6: The NRF generates a new access token, the business ID in the token identifier remains unchanged, the process number is added one, the corresponding business process number stored in the hash table is also modified simultaneously, and the newly generated access token is issued to the Consumer.

If the business process number stored in the hash table does not match the business process number of the token, the request message from the NF can be considered abnormal. Then the NRF returns a message to the Consumer refusing to issue an access token.

3.3.3. The Validation of the Token Request Message

When the NRF receives the message GAT from the Consumer requesting an access token, it will validate the request message from the Consumer. The NRF verifies that the service is authorized and verifies the integrity of the access token. The UE and business information shall remain unchanged in the same business procedure.

The NRF checks whether the Consumer is authorized to access the requested service.

if F1(CONS||PROS) = AUTH,

if RAT.CLA = RV_Kp′(RAT.RIA),

If PAT.CLA = RV_Kn(PAT.RIA),

if GAT.UID = PAT.CLA.UID′,

if GAT.5GCPP = PAT.CLA.5GCPP′,

if F2(5GCPP′||CONS′||PROS′||REQS′) = (CONS||PROS||REQS),

if F3(5GCPP′||PID′) = PAT.CLA.JTI′.EID′.

The NRF queries the dynamic access token repository to verify that the requested service corresponds to the latest business process. The NRF issues a new access token AT to the authenticated Consumer and signs the token. A unique token identifier is generated, and the corresponding process serial number stored in the access token repository is updated.

JTI. PID = JTI′.PID′,
JTI. EID = JTI′.EID′ + 1,
MODA(5GCPP||PID||EID),
JTI = 5GCPP||PID||EID,
AT.CLA = CONS||PROS||5GCPP||UID||REQS||Ta||Te||JTI,
AT.RIA = (CONS||PROS||5GCPP||UID||REQS||Ta||Te||JTI) Kns,
AT = AT.CLA||AT.RIA.

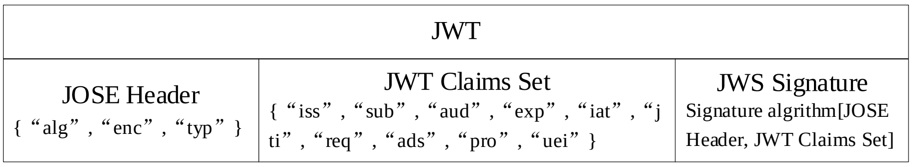
3.4. The NRF Issues the Access Token

3.4.1. JSON Web Token (JWT)

A JWT comprises a JOSE Header, a JWT Claims Set, and a JWS Signature [23]. The core content of JWT is a set of claims represented as JSON objects, which consist of multiple sets of name-value pairs. The JOSE Header declares the type of the encoded object, the signature algorithm, and the encryption algorithm, and the "cty" header parameter declares that the token contains nested tokens. JWT Claims Set consists of claims that are the body object passed by JWT with a string name and an arbitrary JSON value.

The claim names in the JWT Claims set must be unique, and there are three types of claim names: registration, public, and private. The terms of the registration claims refer to seven claims defined in the protocol that specifies the purpose of the claim, including "iss", "sub", "aud", "exp", "nbf", "iat", and "jti". The JWT user can freely define the public claim name according to actual needs. The user specifies the name of the claim and describes the object, and the claim name can be used publicly in the valid space of the JWT.

The JOSE Header and JWT Claims Set in the ASCII representation are signed using the signature algorithm declared in the header argument to get the JWS Signature, as shown in Figure 7. Base64url-encoded JOSE Header, JWT Claims Set, and JWS Signature are separated by "." Connect, then we get a JWT.



JWT		
JOSE Header { "alg" , "enc" , "typ" }	JWT Claims Set { "iss" , "sub" , "aud" , "exp" , "iat" , "jti" , "req" , "ads" , "pro" , "uei" }	JWS Signature Signature algrithm[JOSE Header, JWT Claims Set]

Figure 7. JWT structure.

3.4.2. Access Token Structure

The NRF can write UE information, business information, and authorization scope information to the token. Through the access token, the NRF passes the authorization information to the Producer, who verifies whether the service requested by the Consumer is authorized to be accessed based on the authorization scope of the access token.

Based on RFC7519 protocol, add registration claim names "iat" and "jti" and public claim names "req", "ads", "pro" and "uei" to the token claim set. The "iat" claim identifies the time of issuance of the token. The "jti" statement provides a unique identifier for the token, both to prevent it from being replayed and to help the NF manage the token more quickly and efficiently. The claim name of the public claim and the role of the claim are defined according to the actual needs of the current improved service access process for accessing tokens. The "req" claim identifies the service requested by the Consumer. The "ads" identifies the additional scope of the token, where the specific service operations that the Consumer is authorized to access can be written into the "additional scope", as defined by the 3GPP protocol [19]. The "pro" claim identifies the 5GC service to which the Consumer service request belongs. The "uei" claim identifies the identity of the UE served by the Consumer. The description of the claims is shown in Table 5.

Table 5. The content declared in the token.

Claim Name	Role	Example
iss	Identifies the body that publishes the JWT	NRF_01
sub	Identifies the usage subject of the JWT	AMF_01
aud	Identifies the target audience of the JWT	SMF
exp	Identifies the expiration time of the JWT	2022.1200.1200
iat	Identifies the issue time of the JWT	2022.1200.1155
jti	Provide a unique identifier for the JWT	00010000001202
req	Identifies the authorized services of the JWT	Nsmf_PDUSession
ads	Identifies the additional scope of the JWT	Nsmf_PDUSession_ CreateSMContext
pro	Identifies the business of the JWT subject	PDU Session Establishment
uei	Identifies the UE information for the subject served	SUCI-02087500012345678

3.5. The Producer Validates the Service Request for NF

3.5.1. The Producer Shares the Responded Access Token Repository

In the current service access procedure, the "exp" claim in the access token issued by NRF specifies the validity period of the token. To ensure the smooth completion of the service, the validity period of the token restriction is relatively loose. The Consumer can repeatedly initiate access to the Producer during the validity period of the token. When the audience claim of the token does not specify a specific Producer, the Consumer can request services from a group of Producers that match the audience claim after obtaining the token, which brings a hidden danger to the protection of UE privacy data. In the above procedure, by adding the UE information field in the token, the service access scope of the NF can be restricted more clearly and effectively. Still, it can't solve the security risk caused by the NF holding the token for repeated access.

To limit the number of accesses of Consumers holding tokens, this paper proposes to establish a hash table to store the repository of the responded access tokens. A group of Producers of the same type shares a responded access token repository. At this point, the hash table does not have to consider anti-collision. If multiple token identifiers point to the same storage space, the corresponding different tokens can be stored together in this storage space. The specific implementation and usage settings of the shared token repository are as follows.

Token Authentication:

1. When the Producer receives an access request from a Consumer holding a token, it first finds out whether the token is stored in the address space corresponding to the identifier in the shared token database. The token identifier is a fourteen-digit decimal number, and the Producer hashes the token identifier and uses the hash result as the storage address of the token.
2. If the re-signature token is not stored in the address, or the token identifier stored in the address space is inconsistent with the token identifier received by the Producer, the token is first used for access.
3. If the token stored in the address space has the same token identifier as the token received by the Producer, the token has already been used. The Producer will reject the access request carrying this token.

Token Storage:

When the Producer responds to a service request from a Consumer, it re-signs the token and writes the token identifier to the shared token repository. The Producer adds two new characters to the token identifier to identify the Producer and stores the re-signed token together with the shared token repository.

By sharing the repository of responded access tokens, a group of Producers can share the usage conditions of access tokens and specify the identity of the Producer who responded to the token.

3.5.2. The Producer Re-Signs the Token

In a business procedure, the triggering of the NF services is timed. An NF triggers a request to invoke the next NF service only when its service is invoked or receives a response from another NF. Then the Consumer of the following process must be the Producer/Consumer of the current process. The Producer re-signs the access token and returns it to the Consumer after the request message is validated. The re-signature mechanism ensures that the Consumer of the following process holds the re-signature access token of the previous process.

There are three purposes for designing the re-signature token of the Producers: The first is to declare that the service request has been executed by the Producer. The second purpose is to identify the Producer so that NRF can verify the authenticity of the Consumer request message. The third purpose is to transfer the associated information of the current process to NRF so that both the UE identity information and business information remain untouched throughout the entire business procedure.

The Producer re-signs the access token in the request message, generating a new nested token. The issuing time of the token declares when the current process starts to respond. The token identifier follows the identifier of the access token issued by the NRF. And the Producer uses its private key to sign the declaration content.

The Consumer must carry the re-signature token of the previous process in the message requesting the access token. If the previous process Producer information carried in the request message is incorrect or the token has been tampered with, the NRF will fail to verify the token signature. At this time, the NRF will return a message error response to the Consumer and refuse to issue an access token to the Consumer.

3.5.3. The Producer's Response to the Request

When the Producer receives an access request carrying a token, the Producer processes the request message in the following steps.

Step 1: The Consumer requests a service from the Producer.

$\mathrm{RSQM} = \mathrm{CONS} \,||\, \mathrm{PROS} \,||\, \mathrm{5GCPP} \,||\, \mathrm{UID} \,||\, \mathrm{REQS} \,||\, \mathrm{AT}$.

Step 2: The Producer verifies the integrity of the token and verifies that the token is for initial use.

$\mathrm{RV\_Kn(AT.RIA)} = \mathrm{AT.CLA} = \mathrm{CONS} \,||\, \mathrm{PROS} \,||\, \mathrm{5GCPP} \,||\, \mathrm{UID} \,||\, \mathrm{REQS} \,||\, \mathrm{Ta} \,||\, \mathrm{Te} \,||\, \mathrm{JTI}$, if $\mathrm{F4(JTI)} = \mathrm{NIL}$.

Step 3: The Producer verifies that the declared target audience of the token matches its information.

if $\mathrm{NFID} \in \mathrm{AT.CLA.PROS}$.

Step 4: The Producer validates the request message based on the claims set of the token.

if $\mathrm{RSQM.CONS} = \mathrm{AT.CLA.CONS}$,

if $\mathrm{RSQM.5GCPP} = \mathrm{AT.CLA.5GCPP}$,

if $\mathrm{RSQM.UID} = \mathrm{AT.CLA.UID}$,

if $\mathrm{RSQM.\ REQS} \in \mathrm{AT.CLA.REQS}$,

if $\mathrm{RSQM.\ RT} \in (\mathrm{AT.CLA.Ta}, \mathrm{AT.CLA.Te})$.

Step 5: The Producer re-signs the token.

$\mathrm{RAT} = \mathrm{AT} \,||\, \mathrm{(AT)Kps}$,

$\mathrm{MODB(JTI)} = \mathrm{RAT} \,||\, \mathrm{NFID}$.

Step 6: The Producer performs the service requested by the Consumer and returns a response to the Consumer.

$\mathrm{RESP} = \mathrm{Kc}\{\mathrm{EXE(REQS)} \,||\, \mathrm{RAT}\}$.

3.6. The Initiation of Business Procedures in 5GC

At the starting point of the business procedure within 5GC, the service request initiated by the NF is the first process of the business. Currently, the NF is requesting an access token from the NRF with the following message.

GAT = Kn(CONS||PROS||5GCPP||UID||REQS).

After completing mutual authentication with the Consumer, the NRF looks in the static business process chart to see if the information of the first process of the business matches the information in the Consumer request message.

if GAT.(CONS||PROS||REQS) = F2(5GCPP).

The NRF creates a new number PID and token identifier 5GCPP || PID || EID for the business initiated by this request. Then, the NRF updates the actual business number of the business in the dynamic access token repository.

GET(5GCPP) = PID,
New PID = PID + 1,
EID = 01.

The generated token identifier is 5GCPP||New PID||EID. The NRF generates the access token while updating the PID in the token allocation table, and stores the token identifier of the newly issued access token in the address corresponding to the hash table.

JTI = 5GCPP||New PID||EID,
MODC(5GCPP||New PID||EID),
AT.CLA = CONS||PROS||5GCPP||UID||REQS||Ta||Te||JTI,
AT.RIA = (CONS||PROS||5GCPP||UID||REQS||Ta||Te||JTI) Kns,
AT = AT.CLA||AT.RIA.

4. Analysis and Verification

4.1. Non-Formal Analysis

The existing service access process is shown in Figure 1. Taking the third step of the UE authentication process shown in Figure 2 as an example, we analyze the improved service access procedure as shown in Table 6. Currently, the business process is 02. Assume that the Consumer is AUSF_1 and the instance ID is AUSF_01, the NRF instance is NRF_1, and the instance ID is NRF_01. The set of a group of UDMs is S_{UDM} = {UDM_1, UDM_2, UDM_3}. Assume that the UE ID space is U, and the service ID space is P.

Assume that the UE identity information requested for authentication is $a \in U$, and the service ID assigned by the current authentication service is $x \in P$.

Table 6. The proocessing flows of the improved NF service access procedure.

1. The Consumer requests a token from the NRF
Message body: {"consumer type": "AUSF"; "instance id": "AUSF_01"; "expected NF type": "UDM"; "expected NF service name": "Nudm_UEAuthentication_Get"; "UEID": "a"; "procedure ID": "x"}. Previous access token Claims Set: {"iss": "NRF_01"; "sub": "AMF_01"; "aud": "AUSF"; "req": "Nausf_UEAuthentication Service"; "ads": "Nausf_UEAuthentication_Authenticate"; "uei": "a"; "pro": "x"; "jti": "x,01"; "iat": "202210110911"; "exp": "202210110915"}. Signature algorithm: The private key of AUSF.
2. The NRF validates the request message
2.1 Check if AUSF is authorized to access UDM. 2.2 Check whether the information in the request message is consistent with that in the token. 2.3 The storage address of the access token is y. Check whether the "jti" of the token stored in the storage space corresponding to address y is the same as the "jti" in the request message. 2.4 Generate an access token. Update the "jti" stored at address y in the token repository.
3. The Consumer requests services from the Producer
Service Request: {"consumer type": "AUSF"; "instance id": "AUSF_01"; "expected NF type": "UDM"; "expected NF service name": "Nudm_UEAuthentication_Get"; "UEID": "a"; "procedure ID": "x"; "Access token"}
4. The Producer authentication token claims
4.1 Ensure the integrity of the token. 4.2 Check whether the audience claim in the access token matches its own identity or the type of the Producer. 4.3 Check the expiration time in the access token against the current time. 4.4 Check whether the responded token repository contains this token. 4.5 Check whether the information in the request message matches the corresponding information in the token. 4.6 Checks whether the authorized scope matches the requested service operation.
5. The Producer's response to service requests
5.1 Response request message. 5.2 Store the access token id in the repository and generate a new token for nested access tokens. 5.3 Response to the Consumer with the new token and finish the service.

This scheme is analyzed from a theoretical design perspective.

1. Mutual Authentication: TLS encryption mechanism at the transport layer during communication between NFs ensures that the identity of both NFs is accurate and trustworthy.
2. Logic Reasonableness: NRF adds the token identifier embedded with the business process to the access token to ensure the orderly occurrence of NF service requests and realize the verification of the rationality of the business logic of service access requests.
3. Non-repudiation: NF signed token is non-modifiable. The NRF authorizes the Consumer to access and issues the token, the Producer re-signs the token and returns it to the Consumer at the end of the response, and the response messages of the NRF and the Producer have non-repudiation. The Consumer's access request must carry an access token and its identity information, and the token states the Consumer's identity information. The message receiver in the service access procedure matches the request message with the token claims, thus ensuring that the Consumer's request messages also have non-repudiation.
4. Confidentiality: The messages transmitted between NFs are encrypted by public keys to ensure the confidentiality of the transmitted information.
5. Integrity: The access token signed by the private key is tamper-proof. Thus, the token message has integrity. The presence of a token signature and token verification ensures the integrity of the request message.
6. UE Privacy: The improved scheme proposed in this paper ensures that the UE identity information will not be changed during the whole process of business execution. The Consumers cannot access the private data of other UEs at will, ensuring the security of UE privacy data.
7. Anti-replay: The access token issued by the NRF is unique. Based on a shared access token repository, the Consumers can not repeat requests for Producer services with the same token. The NRF stores the latest process of the current business in the access token repository. Thus, the Consumers can not repeat requests for authorization of service.
8. Anti-man-in-the-middle Attack: The communication messages between NFs use a public key encryption mechanism, and only the designated recipient can decrypt the cipher text with the private key. Thus, the man-in-the-middle cannot decrypt the communication messages between NFs in the NF service access procedure and cannot effectively tamper with the messages. Moreover, NFs will conduct mutual authentication before communication to ensure the authenticity of each other's identity, making it difficult for man-in-the-middle attacks to succeed.
9. Anti-overstepping: The NRF-issued access token adds the token identifier, business information, and UE identity information fields. The whole business procedure is chained by the token identifier, and the distinction between different businesses is achieved by assigning a unique business ID to each specific business. Based on the chained access relationship of NFs in the business procedure, the actual service access performed in the procedure is chained by identifying the business procedure of the specific business, which limits the possibility of overstepping access of NFs brought by the NRF authorization mechanism. The UE identity information field added in the token ensures that the UE identity is not changed in the whole process of this service and limits the access or operation rights of the Consumers to the UE information. The Producer can control the access behavior of the NF after obtaining the token by sharing the access token repository and avoid the Consumer from initiating repeated access to multiple Producers or the same Producer with the same token.

4.2. Formal Analysis

AVISPA is a highly automated protocol security validation tool. In recent years, the formal analysis of 5G-related protocols is mostly done by AVISPA [18], ProVerif [13], and Tamarin Prover [15]. Over the years, many scholars have used AVISPA to analyze the security of proposed solutions. AVISPA can simulate Message Sequence Charts (MSC) according to the input protocol flow so that users can intuitively judge whether the abstraction of the protocol is reasonable. AVISPA has a systematic logical language for describing changes in subjects, sessions, parameters, and environments over the course of the protocol. At the same time, AVISPA can achieve two security authentication goals, one is message confidentiality, and the other is mutual authentication between subjects to prevent replay attacks. These two security verification objectives basically meet our requirements for the validity verification of the proposed scheme.

Before AVISPA can be used to validate protocol security, it is necessary to abstractly encode the natural language description of the protocol procedure into a normalized High-Level Protocol Specification Language (HLPSL) form. SPAN automatically converts the intuitive and readable CAS+ language into the HLPSL language [24]. For writing protocol procedures that conform to the HLPSL form of the specification, the SPAN tool can generate (MSC for the protocol).

To make the protocol described in natural language automatable for analysis, it is first necessary to abstract the protocol into intuitively clear steps of the interaction procedure. The states of the interacting parties will change in each step. As shown in Figure 8, a state transition system can be constructed from the protocol flow, which includes the trigger condition, action after the trigger, and the real-time state of the subject before and after the trigger.

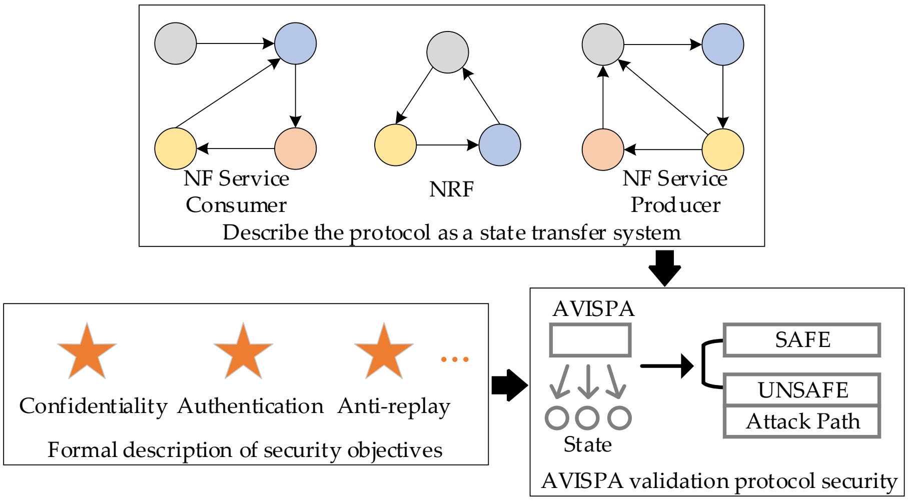

NF Service
Consumer
NRF
NF Service
Producer
Describe the protocol as a state transfer system
Confidentiality
Authentication
Anti-replay
...
Formal description of security objectives
AVISPA
State
SAFE
UNSAFE
Attack Path
AVISPA validation protocol security

Figure 8. A security analysis model of NF service access protocol based on AVISPA.

From the beginning to the end of a service access process, the NF states involved in the process will change several times. We use different colors to distinguish different NF states in Figure 8. The star symbol means different security objects, such as confidentiality, authentication, and anti-replay. AVISPA automatically verifies that the protocol meets its security objectives based on the state changes of NF and the security objectives of validation. If the security target is not met, the output result of AVISPA is “UNSAFE” and the possible attack path of the attacker is displayed.

We verify the security of the existing service access procedure and the enhanced service access procedure through AVISPA. We abstract the current service access procedure protocol specification and the enhanced service access procedure protocol specification into CAS+ language, convert CAS+ language into HLPSL language by the SPAN tool, and then verify the protocol’s security before and after modification through simulation.

4.2.1. Service Access Procedure

As shown in Table 7, the protocol description of CAS+ language is divided into five parts: Identifiers, Message, Knowledge, Session Instance, and Intruder Knowledge.

Table 7. CAS+ description of the NF service access procedure.

Protocol of NF Access Procedure
identifiers
NFA, NFB, NRF : user;
ReqS1, Token1, Resp1 : number;
Ka, Kb, Kn : public_key;
messages
1. NFA -> NRF : {NFA, NFB, ReqS1}Kn
2. NRF -> NFA : {NFA, NFB, ReqS1, Token1}Kn′
3. NFA -> NFB : {NFA, NFB, ReqS1, Token1}Kb
4. NFB -> NFA : {Resp1}Ka
knowledge
NFA : NFA, NFB, NRF, Ka, Kb, Kn, ReqS1;
NFB : NFA, NFB, NRF, Ka, Kb, Kn, Resp1;
NRF : NFA, NFB, NRF, Ka, Kb, Kn, Token1;
session_instances
[NFA:consumer, NFB:producer, NRF:server, Ka:ka, Kb:kb, Kn:kn, ReqS1:reqs1, Token1:token1, Resp1:resp1]
[NFA:i, NFB:producer, NRF:server, Ka:ki, Kb:kb, Kn:kn, ReqS1:reqs1, Token1:token1, Resp1:resp1];
intruder_knowledge
producer, ka, kb, kn, ki, ki′, reqs1, token1;

The first part, Identifiers, declares the parameters involved in the process. The NRF, NFA, and NFB declared as users represent the three entities in the service access procedure, NRF, NF Service Consumer, and NF Service Producer, respectively. Ka, Kb, and Kn declared as public_key are the public keys of NFA, NFB, and NRF, respectively. ReqS1, Token1, and Resp1 represent the service requested by NFA, the access token issued by NRF, and the response of NFB to the service request in the service access procedure, respectively.

The second part, Message, describes the entire interaction between the three entities. The service access procedure can be simplified into four steps: (1) The NFA requests the access token to the NFB from the NRF; (2) The NRF generates an access token and returns it to the NFA; (3) The NFA requests service from the NFB with token; (4) The NFB responds to the NFA request.

The third part, Knowledge, is the information that each entity has mastered before the service access procedure begins. After the procedure starts, the entity generates interactive messages based on the available data.

The fourth part, Session Instance, assigns specific values to identifiers defined in the declaration. In the interaction between NFs, the values represented by identifiers are passed. A session instance usually contains at least two sessions, one is a regular session, and the other is an attacked session. The second session in Table 7, "NFA:i" indicates that the entity that initiated the session, NFA, is the attacker.

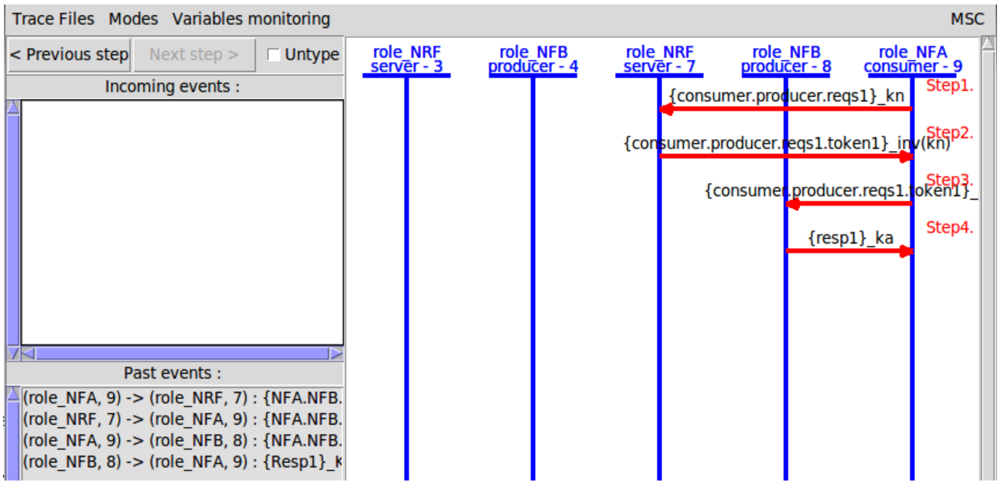
The fifth part, Intruder Knowledge, is the knowledge known to the attacker in the attacked session instance before launching the attack. As shown in Table 7, the intruder knows the identity of the Producer and the public keys of NFA, NFB, and NRF.

The sixth part is the Goal. As shown in Table 8, the security target of authentication is set in HPLSL to verify the confidentiality, identity authentication, and anti-replay of parameters in the protocol procedure.

Table 8. Verification objectives of formal analysis.

Verification Goal
goal
secrecy_of reqs1, token1, resp1
authentication_on server_consumer_reqs1
authentication_on consumer_producer_resp1
end goal

As shown in Figure 9, a message sequence diagram of the current service access procedure can be obtained by the SPAN tool.

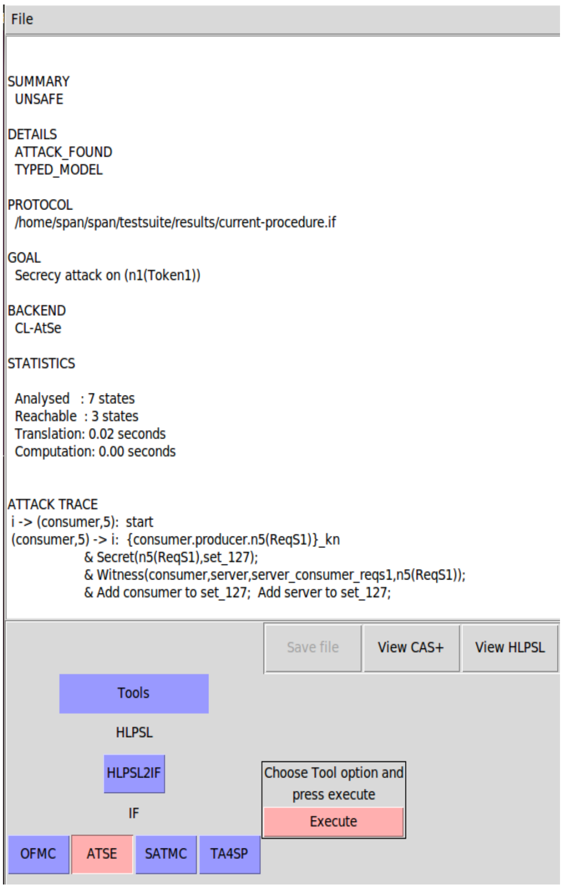


```
{consumer.producer.reqs1}_kn
{consumer.producer.reqs1.token1}_inv(kn)
{consumer.producer.reqs1.token1}_
{resp1}_ka
```

```
(role_NFA, 9) -> (role_NRF, 7) : {NFA.NFB.
(role_NRF, 7) -> (role_NFA, 9) : {NFA.NFB.
(role_NFA, 9) -> (role_NFB, 8) : {NFA.NFB.
(role_NFB, 8) -> (role_NFA, 9) : {Resp1}_k
```

Figure 9. MSC of SPAN simulating service access procedure.

Perform Attack-Searcher (ATSE) security verification on the protocol procedure. The background analysis algorithm used by ATSE is Constraint Logic-based Attack-Searcher (CL-AtSe), constraint logical-based attack-searcher. CL-AtSe is built based on modeling, the simplified probe method, and the redundancy elimination technique. Then the protocol errors can be found, and the finite session can be verified. The validation results of the current service access procedure are shown in Figure 10.



```
SUMMARY
 UNSAFE

DETAILS
 ATTACK_FOUND
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/current-procedure.if

GOAL
 Secrecy attack on (n1(Token1))

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 7 states
 Reachable  : 3 states
 Translation: 0.02 seconds
 Computation: 0.00 seconds


ATTACK TRACE
 i -> (consumer,5):  start
 (consumer,5) -> i:  {consumer.producer.n5(ReqS1)}_kn
                & Secret(n5(ReqS1),set_127);
                & Witness(consumer,server,server_consumer_reqs1,n5(ReqS1));
                & Add consumer to set_127;  Add server to set_127;
```

Figure 10. Formal analysis results of the NF service access procedure.

SUMMARY is the overall summary of the protocol validation. DETAILS is a detailed explanation of the validation result. PROTOCOL is the path information of the validated protocol file. GOAL is the security target of the protocol validation. BACKEND is the background analysis algorithm of the protocol security. STATISTIC is the specific validation details of the validation algorithm. ATTACK TRACE is the attack path.

The result of SUMMARY is SAFE or UNSAFE. When the result of SUMMARY is SAFE, the protocol is safe. When the result of SUMMARY is UNSAFE, the protocol may be attacked by attackers, DETAILS will show ATTACK_FOUND, and ATTACK TRACE will show the specific attack path. In this paper, we use ATSE to verify the security of the current protocol and the improved protocol. The verification result of the present service access procedure is "UNSAFE", and Figure 10 shows the attack path of the attacker. The attack path shows that in the current protocol procedure, the attacker can launch a replay attack to NFB disguised as a legitimate NF by eavesdropping and forwarding the messages of inter-NF communication.

4.2.2. Advanced Service Access Procedure

The improved service access process CAS+ form of this paper is described in Table 9.

Table 9. CAS+ description of the advanced service access procedure.

Protocol of Advanced NF Access Procedure
identifiers
NFA, NFB, NFC, NRF : user;
ReqS1, Token1, Resp1, ReqS2, Token2, Resp2 : number;
Ka, Kb, Kn, Kc : public_key;
Hash : hash_func;
messages
1. NFA -> NRF : {NFA, NFB, ReqS1}Kn
2. NRF -> NFA : {NFA, NFB, ReqS1, Token1}Kn′
3. NFA -> NFB : {NFA, NFB, ReqS1, Token1}Kb
4. NFB -> NFA : {NFA, NFB, Resp1, Token1}Ka
5. NFB -> NRF : {NFB, NFC, ReqS2, Token1}Kn
6. NRF -> NFB : {NFB, NFC, ReqS2, Token2}Kn′
7. NFB -> NFC : {NFB, NFC, ReqS2, Token2}Kc
8. NFC -> NFB : {NFB, NFC, Resp2, Token2}Kb
knowledge
NFA : NFA, NFB, NFC, NRF, Ka, Kb, Kc, Kn, Hash;
NFB : NFA, NFB, NFC, NRF, Ka, Kb, Kc, Kn, Hash;
NFC : NFA, NFB, NFC, NRF, Ka, Kb, Kc, Kn;
NRF : NFA, NFB, NFC, NRF, Ka, Kb, Kc, Kn, Hash;
session_instances
[NFA:con,NFB:pro1,NRF:server,NFC:pro2,Ka:ka,Kb:kb,Kc:kc,Kn:kn,Hash:h]
[NFA:con,NFB:i,NRF:server,NFC:pro2,Ka:ka, Kb:ki, Kc:kc,Kn:kn,Hash:h];
intruder_knowledge
con, server, pro2, kc, ka, kb, kn, ki, ki′;

After converting CAS+ language into HLPSL language form, HLPSL is debugged to increase security targets. The protocol described in HLPSL form is divided into four parts: Role, Session, Environment, and Goal.

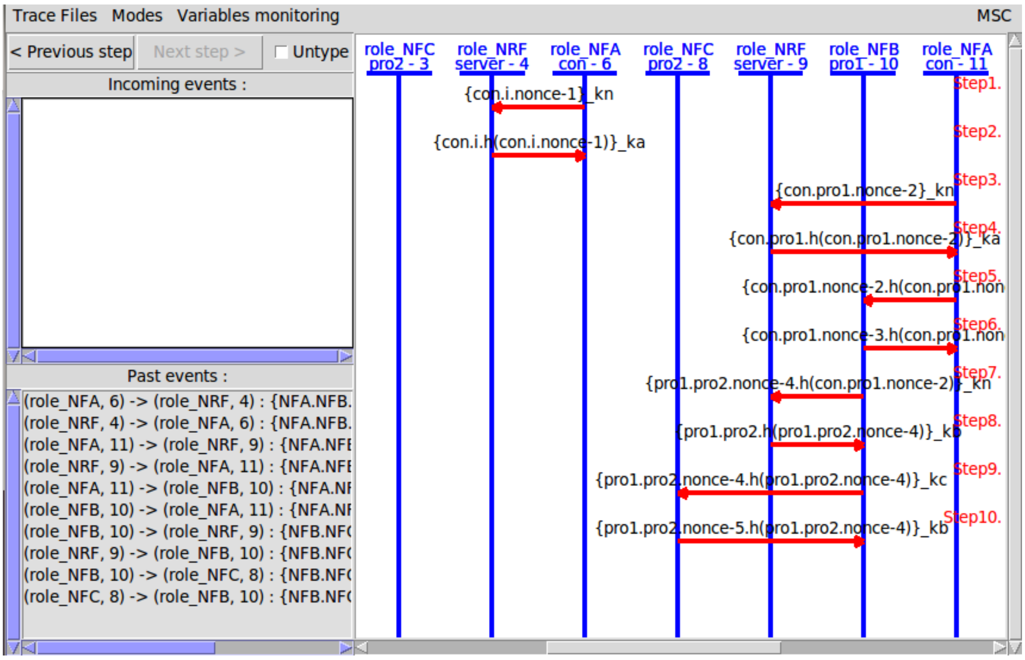
The first part, Role, includes the parameters and state transitions involved in the role. Take the definition of parameters and states transition of the subject NFA as an example, as shown in Table 10. The Role_NRA brackets define the knowledge known to the NFA before the process starts, including the subject NFA, NFB, NFC, and NRF, the public keys Ka, Kb, Kc, and Kn corresponding to the subject, the hash function, and the sending channel SND and receiving channel RCV. Local defines the new parameters generated or received during the protocol procedure, including the subject state, State, request message, ReqS1, response message, Resp1, and access token, Token1. The HLPSL descriptions of the NFB, NFC, and NRF are similar to those of the NFA.

The second part, Session Definition, completely describes the parameters and subjects involved in a session instance. The four entities described in this experiment are NFA, NFB, NFC, and NRF, as shown in Table 10. The general knowledge of each subject is also defined in the session section.

The third part, Environment, assigns actual values to all parameters used in the procedure and passes the values corresponding to the parameters to the actual session through Session. At this time, the environment defines the knowledge of the intruder.

The fourth part, Goal, describes the security goal of the protocol procedure that needs to be verified by the background analysis algorithm. The Role section defines the secret encryption content, and the Goal section uses "Securecy_of" to verify the security of the encryption parameter. "Witness" and "Request" in the subject definition are bound predicate relations, which are target facts related to authentication, used to check whether a subject is correct, whether the target peer is believed to exist in the current session, and whether both entities in the session agree on a certain parameter and this parameter is not allowed to be replayed. "Authentication_on" is the mutual authentication between subjects and verifying the checked variables. We want to ensure the confidentiality of communication messages between NFs and prevent replay attacks on messages. The security target for the transmission parameters is set in the Goal section, as shown in Table 10.

The MSC of the improved protocol procedure is shown in Figure 11, and the Formal analysis results are shown in Figure 12. The improved scheme verifies as secure under a limited number of session instances. Both parameter security and inter-entity authentication are verified to be safe and valid. After introducing the token identifier, the re-signature mechanism, and the shared token repository into the service access procedure, the attacker cannot launch a replay attack on the Producer.



```
(role_NFA, 6) -> (role_NRF, 4) : {NFA.NFB.
(role_NRF, 4) -> (role_NFA, 6) : {NFA.NFB.
(role_NFA, 11) -> (role_NRF, 9) : {NFA.NFE
(role_NRF, 9) -> (role_NFA, 11) : {NFA.NFE
(role_NFA, 11) -> (role_NFB, 10) : {NFA.NF
(role_NFB, 10) -> (role_NFA, 11) : {NFA.NF
(role_NFB, 10) -> (role_NRF, 9) : {NFB.NFC
(role_NRF, 9) -> (role_NFB, 10) : {NFB.NFC
(role_NFB, 10) -> (role_NFC, 8) : {NFB.NFC
(role_NFC, 8) -> (role_NFB, 10) : {NFB.NFC
```

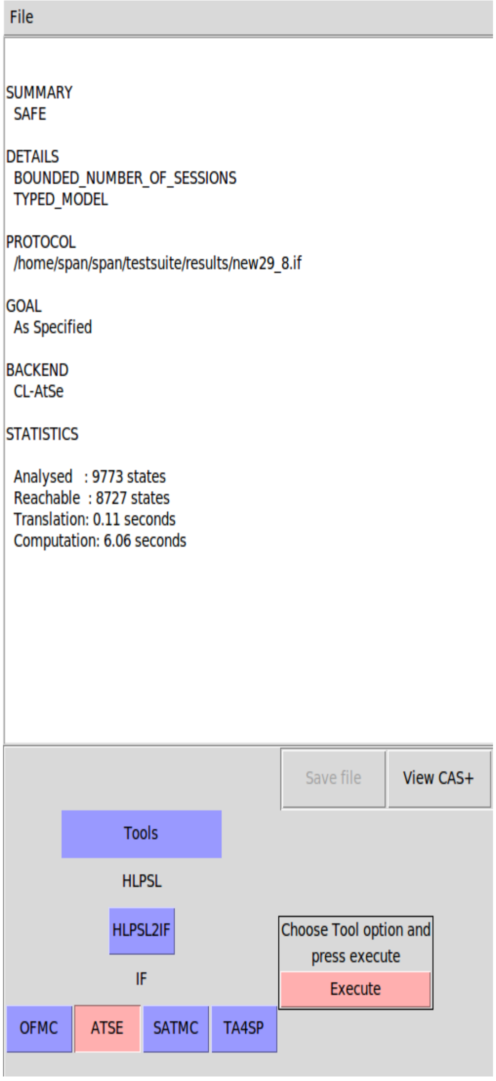
Step1. {con.i.nonce-1}_kn
Step2. {con.i.h(con.i.nonce-1)}_ka
Step3. {con.pro1.nonce-2}_kn
Step4. {con.pro1.h(con.pro1.nonce-2)}_ka
Step5. {con.pro1.nonce-2.h(con.pro1.non
Step6. {con.pro1.nonce-3.h(con.pro1.non
Step7. {pro1.pro2.nonce-4.h(con.pro1.nonce-2)}_kn
Step8. {pro1.pro2.h(pro1.pro2.nonce-4)}_kb
Step9. {pro1.pro2.nonce-4.h(pro1.pro2.nonce-4)}_kc
Step10. {pro1.pro2.nonce-5.h(pro1.pro2.nonce-4)}_kb

Figure 11. The MSC of the advanced protocol procedure.

Table 10. Description of Advanced NF Access Procedure in HPLML form.

Description of Advanced NF Access Procedure in HPLML Form

```
                              Description of NFA
role role_NFA(NFA, NFB, NFC, NRF : agent,
        Ka, Kb, Kc, Kn : public_key,
        Hash : hash_func,
        SND, RCV : channel(dy))
played_by NFA
def=
    local
        State : nat,
        ReqS1, Resp1 : text,
        Token1 : hash(agent.agent.text)
    init
        State := 0
    transition
        0. State=0 /\ RCV(start) =|>
        State':=1 /\ ReqS1' := new()
        /\ SND({NFA.NFB.ReqS1'}_Kn)
        /\ secret(ReqS1',reqs1,{NFA,NRF,NFB})
        3. State=1 /\ RCV({NFA.NFB.Token1'}_Ka) =|>
        State':=2 /\ SND({NFA.NFB.ReqS1.Token1'}_Kb)
        /\ witness(NFA,NFB,pro1_con_reqs1, ReqS1)
        4. State=2 /\ RCV({NFA.NFB.Resp1'.Token1'}_Ka) =|>
        State':=3 /\ request(NFA, NFB, con_pro1_resp1, Resp1')
end role
                         Description of session information
role session(NFA, NFB, NFC, NRF : agent,
        Hash : hash_func,
        Ka, Kb, Kc, Kn : public_key)
def=
    local
        SND4,RCV4,SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
    composition
        role_NFC(NFA,NFB,NFC,NRF,Ka,Kb,Kc,Kn,SND4,RCV4)
        /\ role_NRF(NFA,NFB,NFC,NRF,Ka,Kb,Kc,Kn,Hash,SND3,RCV3)
        /\ role_NFB(NFA,NFB,NFC,NRF,Ka,Kb,Kc,Kn,Hash,SND2,RCV2)
        /\ role_NFA(NFA,NFB,NFC,NRF,Ka,Kb,Kc,Kn,Hash,SND1,RCV1)
end role
                          Description of parameter values
role environment()
def=
    const
        reqs1, token1, resp1, reqs2, token2, resp2,
        pro1_con_reqs1, con_pro1_resp1 : protocol_id,
        ka, kb, kc, kn, ki : public_key,
        h : hash_func,
        server, con, pro1, pro2 : agent
    intruder_knowledge = {con,server,pro2,kc,ka,kb,kn,ki,inv(ki)}
    composition
        session(con,i,pro2,server,h,ka,ki,kc,kn)
        /\ session(con,pro1,pro2,server,h,ka,kb,kc,kn)
end role
                         Description of verification goals
goal
    secrecy_of reqs1, reqs2, token1, token2, resp1, resp2
    authentication_on pro1_con_reqs1
    authentication_on con_pro1_resp1
end goal
```



```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/new29_8.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 9773 states
 Reachable  : 8727 states
 Translation: 0.11 seconds
 Computation: 6.06 seconds
```

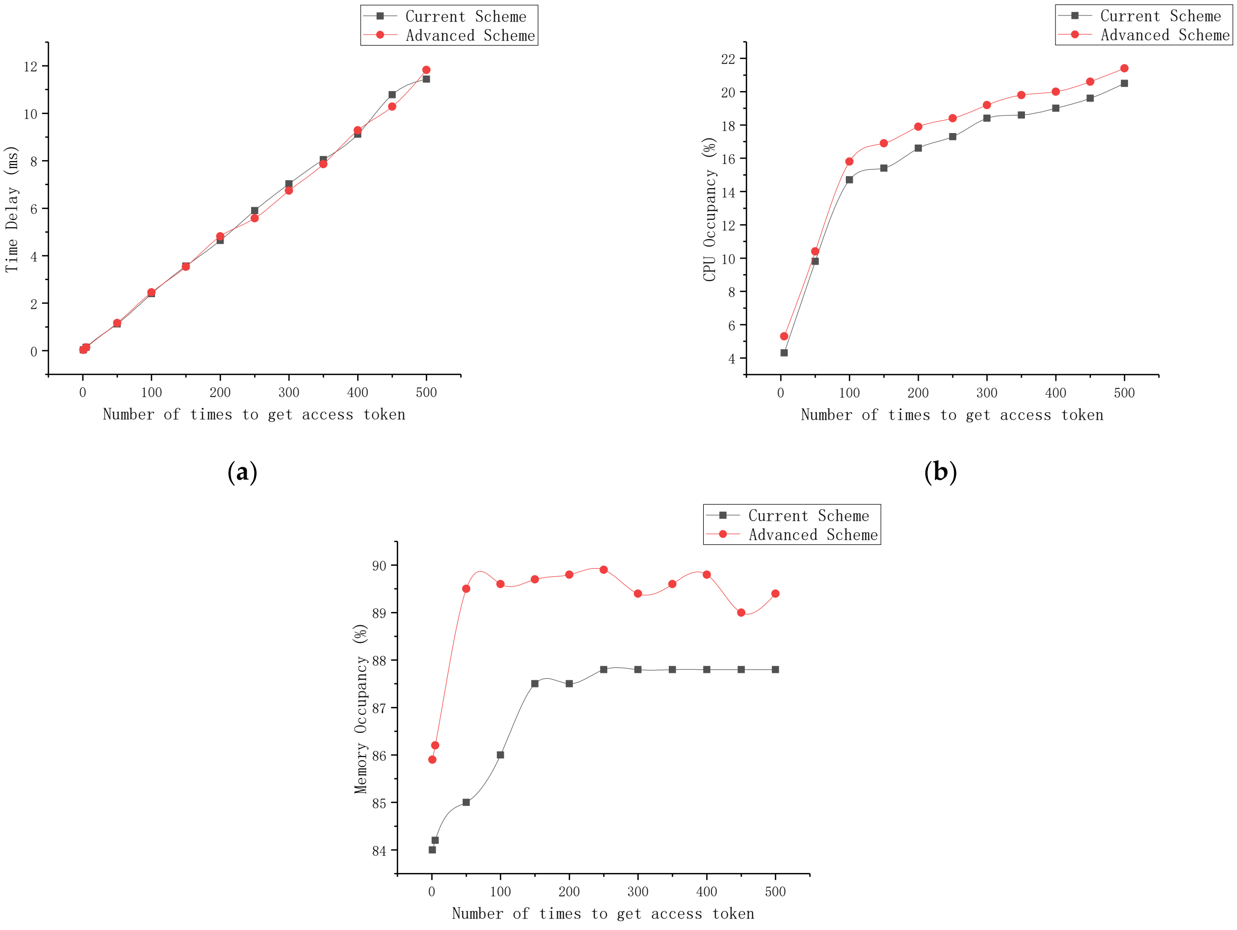
Figure 12. Formal analysis results of the advanced NF service access procedure.

4.3. Performance Evaluation Experiment

This experiment deploys Free5GC on a server with Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz, 4GB RAM, and Ubuntu 18.04 OS. Python3.6 is used as the development environment to test the computation and storage overheads of the existing service access scheme and the improved service access scheme. The latency and CPU usage are used as metrics to measure the performance of the enhanced scheme [4].

In the improved service access procedure, the access token adds some fields, the NRF adds a module for storing the token in a hash table and verifies the token in the request message, and the Producer adds the shared token repository and the re-signature mechanism. These added mechanisms introduce additional computational and storage overheads. Experiments test the latency, CPU occupancy, and memory occupancy from the time the Consumer requests an access token from the NRF to the time the Consumer receives the response from the Producer.

Performance evaluation experiment results are shown in Figure 13. The CPU occupancy and memory occupancy fluctuate slightly with changes in the server environment. The delay of both the current procedure and the improved procedure increases with the number of tokens accessed. The hash function calculation process is added in the improved procedure, so compared with the current procedure, the delay curve and memory occupancy curve of the improved procedure fluctuates more obviously. Meanwhile, we found that sometimes the delay of the improved scheme is smaller than that of the original scheme, but considering that in the process of obtaining tokens, the improved scheme mainly adds a step of the hash calculation, the time complexity is O(1), and the time unit of the delay is ms, which is a small unit. We believe that the reduced delay of the improved scheme may be caused by some errors caused by the state of the computer itself when the program is running, while the overall delay trend of the original scheme is close to linear growth, which is consistent with our expected experimental results. In general, the CPU occupancy increases with the number of tokens acquired by the NF, and the CPU occupancy of the improved procedure increases by at most 12% in the experimental environment used in this paper. The memory occupancy is basically stable at 88%, and the memory occupancy of the improved procedure increases by 2.27% on average. The performance loss of the enhanced scheme is slight and brings negligible efficiency decay. Using hash tables to store token identifiers will consume relatively more storage and computational resources. Still, the increased resource overhead is limited, which is a low-cost mechanism to enhance security.



(**a**)

(**b**)

(**c**)

Figure 13. Diagram of performance evaluation experiment results. (**a**) Time Delay. (**b**) CPU Occupancy. (**c**) Memory Occupancy.

5. Conclusions

5GC sequentially triggers service accesses from different NFs when executing the businesses requested by the UE. This paper analyzes the current NF service access procedure and finds that: (1) The NRF does not verify whether the service requested by the NF matches its business process; (2) Consumers can request access tokens for authorized services from the NRF at any time; (3) Consumers can repeatedly access the services of the Producer after obtaining access tokens. NF services and resources are vulnerable to unauthorized access and replay attacks. To address this problem, this paper proposes a security enhancement scheme based on the current NF service access procedure to achieve management of 5GC business processes and strictly control the scope of services that can be requested by the Consumers. Through the token re-signature mechanism of the Producer, we ensure that 5GC always has an access token with verifiable integrity in delivering the current business information and UE identity information when executing the business. By binding the business process to the token identifier, the NRF can verify the business logic rationality of the Consumer's service request when it receives a request message from the Consumer to obtain an access token. By limiting the number of Consumer accesses to the Producer through a shared access token repository, possible replay attacks can be countered. The security and effectiveness of the enhanced scheme are verified by theoretical analysis and formal analysis tools. The performance evaluation experiments show that the enhanced scheme proposed in this paper brings little efficiency degradation by introducing particular storage overhead and computation overhead.

Author Contributions: Conceptualization, Y.Z. (Ying Zhu) and C.L. (Caixia Liu); methodology, Y.Z. (Ying Zhu) and W.Y. (Wei You); software, Y.Z. (Ying Zhu) and W.Z. (Weicheng Zhang); validation, Y.Z. (Ying Zhu) and Y.Z. (Yiming Zhang); formal analysis, Y.Z. and W.Z. (Weicheng Zhang); investigation, Y.Z. (Ying Zhu) and C.L. (Caixia Liu); resources, Y.Z. (Ying Zhu) and C.L. (Caixia Liu); data curation, Y.Z. (Ying Zhu); writing—original draft preparation, Y.Z. (Ying Zhu); writing—review and editing, Y.Z. (Ying Zhu); visualization, Y.Z. (Ying Zhu) and Y.Z. (Yiming Zhang); supervision, Y.Z. (Yiming Zhang); project administration, C.L. (Caixia Liu); funding acquisition, W.Y. (Wei You). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Major science and technology special Project of Henan Province, grant number 221100210100, Research and Demonstration Application of Security Governance Technology in Network Space in Typical Scenarios.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors thank the reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Kim, H. 5G core network security issues and attack classification from network protocol perspective. *J. Internet Serv. Inf. Secur.* **2020**, *10*, 1–15.
2. Ahmad, I.; Shahabuddin, S.; Kumar, T.; Okwuibe, J.; Gurtov, A.; Ylianttila, M. Security for 5G and beyond. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3682–3722. [CrossRef]
3. Ahmad, I.; Suomalainen, J.; Huusko, J. 5 G-Core Network Security. In *Wiley 5G Ref: The Essential 5G Reference Online*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2019; pp. 1–18.
4. Cao, J.; Ma, M.; Li, H.; Ma, R.; Sun, Y.; Yu, P.; Xiong, L. A survey on security aspects for 3GPP 5G networks. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 170–195. [CrossRef]
5. Bhuiyan, T.; Begum, A.; Rahman, S.; Hadid, I. API vulnerabilities: Current status and dependencies. *Int. J. Eng. Technol.* **2018**, *7*, 9–13. [CrossRef]
6. Chatterjee, M.R. Network Security of 5G network. *EasyChair*. 2021. Available online: https://easychair.org/publications/preprint/jdZr (accessed on 21 December 2022).
7. Zhang, S.; Wang, Y.; Zhou, W. Towards secure 5G networks: A Survey. *Comput. Netw.* **2019**, *162*, 106871. [CrossRef]
8. Zhang, Y.; Liu, C.; Liu, S.; Pan, F. SETOKEN: A secure protection mechanism based on service API for 5G network access token. In Proceedings of the 2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT), Sanya, China, 27–29 December 2021; pp. 1139–1143.
9. Prasad, A.R.; Arumugam, S.; Sheeba, B.; Zugenmaier, A. 3GPP 5G security. *J. ICT Stand.* **2018**, *6*, 137–158. [CrossRef]
10. Fang, D.; Qian, Y.; Hu, R.Q. Security for 5G mobile wireless networks. *IEEE Access* **2017**, *6*, 4850–4874. [CrossRef]
11. Ferrag, M.A.; Maglaras, L.; Argyriou, A.; Kosmanos, D.; Janicke, H. Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes. *J. Netw. Comput. Appl.* **2018**, *101*, 55–82. [CrossRef]

12. Arapinis, M.; Mancini, L.; Ritter, E.; Ryan, M.; Golde, N.; Redon, K.; Borgaonkar, R. New privacy issues in mobile telephony: Fix and verification. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, New York, NY, USA, 16–18 October 2012; pp. 205–216.
13. Hussain, S.; Chowdhury, O.; Mehnaz, S.; Bertino, E. LTEInspector: A systematic approach for adversarial testing of 4G LTE. In Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2018, San Diego, CA, USA, 18–21 February 2018.
14. Basin, D.; Dreier, J.; Hirschi, L.; Radomirovic, S.; Sasse, R.; Stettler, V. A formal analysis of 5G authentication. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 15–19 October 2018; pp. 1383–1396.
15. Liu, C.; Hu, X.; Liu, S.; You, W.; Zhao, Y. Security Analysis of 5G Network EAP-AKA′ Protocol Based on Lowe's Taxonomy. *J. Electron. Inf.* **2019**, *41*, 1800–1807.
16. Dhillon, P.K.; Kalra, S. A lightweight biometrics based remote user authentication scheme for IoT services. *J. Inf. Secur. Appl.* **2017**, *34*, 255–270. [CrossRef]
17. Chen, C.L.; Chiang, M.L.; Hsieh, H.C.; Liu, C.C.; Deng, Y.Y. A lightweight mutual authentication with wearable device in location-based mobile edge computing. *Wirel. Pers. Commun.* **2020**, *113*, 575–598. [CrossRef]
18. Zhang, W.; Wei, H.; Liu, S.; Pu, L. Fast handover authentication scheme in 5G mobile edge computing scenarios5G. *J. Netw. Inf. Secur.* **2022**, *8*, 154–168.
19. 3GPP TS 33.501. Security Architecture and Procedures for 5G System [S]. 2022. Available online: https://www.3gpp.org/ftp/Specs/archive/33_series/33.501 (accessed on 22 September 2022).
20. 3GPP TS 33.501. Security Architecture and Procedures for 5G System [S]. 2020. Available online: https://www.3gpp.org/ftp/Specs/archive/33_series/33.501 (accessed on 16 December 2020).
21. 3GPP TS 23.502. Procedures for the 5G System [S]. 2021. Available online: https://www.3gpp.org/ftp/Specs/archive/23_series/23.502 (accessed on 21 December 2022).
22. Alfoudi, A.S.; Newaz, S.S.; Ramlie, R.; Lee, G.M.; Baker, T. Seamless Mobility Management in Heterogeneous 5G Networks: A Coordination Approach among Distributed SDN Controllers. In Proceedings of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 28 April 2019–1 May 2019; pp. 1–6.
23. Jones, M.; Bradley, J.; Sakimura, N. Json web token (jwt) [S]. No. rfc7519. 2015. Available online: https://www.rfc-editor.org/rfc/rfc7519 (accessed on 21 December 2022).
24. Glouche, Y.; Genet, T.; Houssay, E. SPAN: A security protocol animator for AVISPA [R]. *IRISA*. 2008. Available online: https://www.researchgate.net/publication/228356197_A_security_protocol_animator_tool_for_AVISPA (accessed on 21 December 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.