Kim *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:265
https://doi.org/10.1186/s13638-019-1592-0

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

5G wireless P2MP backhaul security protocol: an adaptive approach

Jiyoon Kim, Gaurav Choudhary, Jaejun Heo, Daniel Gerbi Duguma and Ilsun You*

Abstract

5G has introduced various emerging demands for new services and technologies that raised the bar for quality of service, latency, handovers, and data rates. Such diverse and perplexing network requirements bring numerous issues, among which security stands in the first row. The backhaul, which can be implemented as a wired or wireless solution, serves as a bridge between the radio access and core networks assuring connectivity to end users. The recent trends in backhaul usage rely on wireless technologies implemented using point-to-point (PTP) or point-to-multipoint (P2MP) configurations. Unfortunately, due to the nature of the transmission medium, the wireless backhaul is vulnerable and exposed to more various security threats and attacks than the wired one. In order to protect the backhaul, there have been several researches, whose authentication and key exchange scheme mainly depends on the existing security standards such as transport layer security (TLS), Internet Key Exchange version 1 (IKEv1), IKEv2, Host Identity Protocol (HIP), and Authentication and Key Agreement (AKA). However, such security standards cannot completely fulfil the security requirements including security policy update, key update, and balancing between security and efficiency, which are necessary for the emerging 5G networks. This is basically the motive behind why we study and propose a new security protocol for the backhaul link of wireless access network based on P2MP model. The proposed protocol is designed to be 5G-aware, and provides mutual authentication, perfect forward secrecy, confidentiality, integrity, secure key exchange, security policy update, key update, and balancing trade-off between efficiency and security while preventing resource exhaustion attacks. The protocol's correctness is formally verified by the well-known formal security analysis tools: BAN-logic and Scyther. Moreover, the derived lemmas prove that the security requirements are satisfied. Finally, from a comparison analysis, it is shown that the proposed protocol is better than other standard protocols.

Keywords: 5G, Mobile Backhaul, Security protocol, Formal security analysis

1 Introduction

The emerging demands of new services and technologies in the 5G era increase the requirements of quality of service (QoS), low latency, fast handovers, and high data rates. In addition to both the massive number of users and the provision of new services, the amalgamation of such requirements affects the requirements of 5G networks into being excessively diverse. That consequently leads to attackers having more opportunities to exploit different security vulnerabilities in emerging technologies.

Backhaul plays the role of the bridge between the access and core networks while being in charge of the flow of data via supporting reliable communications. Especially in 5G networks, it is expected that backhaul will be used as key technologies to meet the diverse and personalized requirements of new services and technologies that cannot be satisfied by a common network. Backhaul can exist either in the wired or wireless form [1]. A wired solution, often based on leased line or fiber, not only has an expensive implementation cost but also has difficulty to deploy it in remote areas [2]. On the other hand, wireless backhaul solutions are often preferred to overcome these challenges. In more detail, wireless backhaul is easier to install, simpler to extend, and change can be made with lesser effort. Moreover, the use of wireless backhaul is desirable in areas, such as marine or mountainous, where wired facilities are difficult to install [3]. The wireless

*Correspondence: ilsunu@gmail.com
Department of Information Security Engineering, Soonchunhyang University, 31538 Asan-si, Choongchungnam-do, Republic of Korea

backhaul can be implemented as point-to-point (PTP) or point-to-multipoint (P2MP) configurations [4].

In spite of its advantages, the wireless backhaul is more sensitive to the influences of the surrounding environment as compared to the wired backhaul, which makes it more susceptible to different security attacks. Consequently, there have been studies for backhaul security that are intended to mitigate these attacks [5–11]. Even though the existing approaches satisfy the basic security requirements such as confidentiality, integrity, mutual authentication, perfect forward secrecy, and so on, they are failed to provide adaptive security policy and session key updates as well as optimized trade-off between security and efficiency. These requirements are important for the emerging 5G networks and, obviously, vital for the emerging 5G wireless backhaul[12].

Motivated by this, a security protocol is proposed for the 5G wireless backhaul based on P2MP model. The proposed protocol, composed of the initial authentication, key update, and policy update phases, is designed to meet the above basic security requirements while defencing against resource exhaustion attacks. More importantly, it makes the best use of the key update and policy update phases to dynamically adjust the security policy based on the current network situation and the serving network slice's requirements. In this way, the trade-off between security and efficiency can be optimally managed.

The key contributions of this paper are as follows:

- A proposal for a new security protocol for 5G wireless P2MP backhaul
- A formal security analysis of the proposed protocol using BAN-logic and Scyther tool, and
- A detailed comparative analysis between the proposed and existing protocols in terms of security property, computation overhead, and communication overhead.

The remainder of the paper is organized as follows. Section 2 describes the related works about backhaul security and discusses their advantages. In Section 3, we propose a 5G wireless P2MP backhaul security protocol. Then, the security analysis on the proposed protocol is conducted through BAN-logic and Scyther in Section 4. Section 5 presents a comparison analysis and Section 6 concludes the paper.

2 Related work

Backhaul plays a significant role in the process of forwarding packets over global networks. Especially, wireless backhaul has gained remarkable interests as an emerging technology in telecommunication because of its huge throughput speeds, easy deployments, high reliability, and as a wireless last-mile solution the ability to get bandwidth where traditional cable and fiber infrastructure are not available. Moreover, it also has become a desired solution thanks to its various effective features such as reduced operating expenses (OpEx) (by eliminating fiber and leased line costs), cost-effective deployment that easily scales to growing traffic demands, carrier-class reliability, and overcoming challenging environmental barriers to wired network installation.

On the other hand, it is highly possible for network configurations, types of devices, and lack of interoperability to open new opportunities for vulnerabilities and security attacks [11, 13, 14]. Accordingly, without considering such security issues, backhaul network can suffer from the insider, external, and devices attacks such as critical information leakage, privacy breach, DoS attacks, man-in-middle attacks, and so on. Moreover, there are also concerns related to network formations, service migrations and management, and appropriate placements of authentication server. There are various approaches that focus on secure backhaul networks [8, 15]. The main categorizations of backhaul security architecture are trusted domain-based and IPSec virtual private network (VPN)-based. The trusted domain-based architectures count on VPN-based traffic transportations, and provide a different level of service quality for the different traffic types. The trusted domain-based security, although it provides no additional overheads on the backhaul networks, lacks application layer security such as payload encryption, data integrity, and privacy protection. Furthermore, the trusted domain security is also vulnerable to insider attacks, and does not provide a method to shield against active and passive attacks. The IPSec VPN-based backhaul architectures are also based on the VPN-based traffic transportations. The IPSec tunnel uses two types of modes, IPSec tunnel mode and IPSec Bounded End-to-End Tunnel (BEET) mode[16]. The IPSec tunnel mode includes Internet Key Exchange version 2 (IKEv2) [17] and IKEv2 Mobility and Multihoming (MOBIKE) [18] protocols for secure connections. On the other hand, BEET mode uses Host Identity Protocol (HIP) [19]. These modes provide prominent security benefits like authentication and authorization, encryption, privacy protection, and known attack resistance. Moreover, the architectures also support route optimization, load balancing, and fault tolerance. Unfortunately, these architectures lack additional security requirements that are needed to implement security protocols at the backhaul nodes, without affecting performance and quality of services[20].

Some research studies have focused on the potential security risks and solutions with a wireless backhaul network as follows.

Liyanage et al. [15] discussed the backhaul security architectures and presented a performance comparison on the basis of number of message exchanged for secure tunnel establishment of secure backhaul architectures

such as transport layer security (TLS)/secure socket layer (SSL), BEET mode, and tunnel mode. The security of Xhaul networks in terms of privacy and perfect forward secrecy was studied by Sharma et al. [21]. The authors also introduced a key exchange and authentication protocol for handling the security of a mobile terminal which moves across hubs in a 5G scenario generating a mobile Xhaul link network. Sharma et al. [22] presented a security management issues for backhaul- aware vehicle-to-everything (V2X). The authors discussed the existing prominent solutions such as Authentication and Key Agreement (AKA) and Extensible Authentication Protocol (EAP)-AKA with the associated security concerns [23]. They also analyzed on whether it is necessary for the dynamic key update and possibilities of sub-dividing the 5G security functions or not. The HIP-based backhaul security in multi-homed Femtocells was presented by Namal et al. [24]. The HIP provides security with Encapsulated Security Payload (ESP) and supports mobility and security features. The authors further discussed the early solutions femto access points (FAP) authentication and message encryption methods such as EAP-AKA and X.509 certificate. But through various brute force attacks or intrusion, there is a high risk of compromising the authentication token. Liyanage and Gurtov [25] presented two secured VPN architectures for Long-Term Evolution (LTE) backhaul. The one architecture uses IKEv2-based IPSec tunnel mode VPN, and the other architecture is HIP-based IPSec BEET mode VPN. Liyanage et al.[26] analyzed the existing key exchange mechanism such as IKEv1, IKEv2, MOBIKE, and HIP with the IPSec tunnel. The authors also discussed the drawbacks of the existing key exchange mechanisms such as limited access control, less protection against attacks, lack of traffic classification and monitoring facilities, and reliable tunnel establishment procedures.

Note that most of the existing backhaul security approaches mainly depend on TLS, IKEv1, IKEv2, HIP, and AKA as their core authentication and key exchange scheme. These schemes are thus expected to continually serve as a key security solution for the forthcoming 5G networks[27, 28]. However, they cannot completely fulfil the following security requirements, which are important for the emerging 5G networks.

2.1 Security policy update

In 5G networks, there coexist a variety of network slices, each of which has its own unique data traffic and security requirements. That makes each pair of devices (i.e., a hub and a terminal) in a 5G backhaul system have to switch among network slices whenever a slice change happens. Accordingly, the serving backhaul pair of devices should change its security policy (i.e., security strength) including cryptographic algorithms, authentication strength, key size, key lifetime, and so forth based on both the current network situation and the new network slice's security requirements. Also, even in the same network slice, it is needed to adaptively adjust the security level of the serving backhaul pair of devices, which leads to changing the security policy.

2.2 Key update

Once a security policy is decided, the corresponding backhaul pair of devices should periodically update their session keys to decrease the possibility for the key compromise and the information leakage as well as keep its aimed security level. Moreover, such a key update should be efficiently conducted in a way that a back end authentication server is excluded.

2.3 Balancing between security and efficiency

It is important to dynamically keep the best balance between security and efficiency in the 5G backhaul system. Through such best balance, we can avoid the situations where more security is provided than needed at the expense of efficiency, or where efficiency is oversupported at the expense of security.

3 5G wireless P2MP backhaul security protocol

3.1 Preliminary

As shown in Fig. 1, a 5G wireless backhaul system is composed of three entities-terminal, hub, and mobile backhaul authentication server function, which are denoted as TM, HUB, and ASF, respectively. A wireless backhaul path is established between a TM and a HUB, where data traffics are transmitted between the former connected to an access network and the latter connected to their 5G core network. Typically, a HUB is connected to multiple TMs, and a TM and a HUB authenticate each other with the help of their ASF. In more detail, the HUB can depend on its ASF to perform mutual authentication with the TM because it is assumed that every involved TM shares a secret key in advance with the ASF.

At the core side, different components carry out various tasks. In more detail, Access and Mobility management Function (AMF) is responsible for registration, connection, and mobility management; ciphering and integrity protection; and authentication and authorization. User Plane Function (UPF) supports packet forwarding, routing, inspection, etc. While Authentication Server Function (AUSF) performs primary authentication and key exchange with a UE, Security Anchor Function (SEAF) receives an anchor key from the AUSF, which is then used to derive session keys and execute security setups for non-access stratum and access stratum.

The proposed 5G wireless P2MP backhaul security protocol essentially aims to support the security requirements including mutual authentication, confidentiality, integrity,

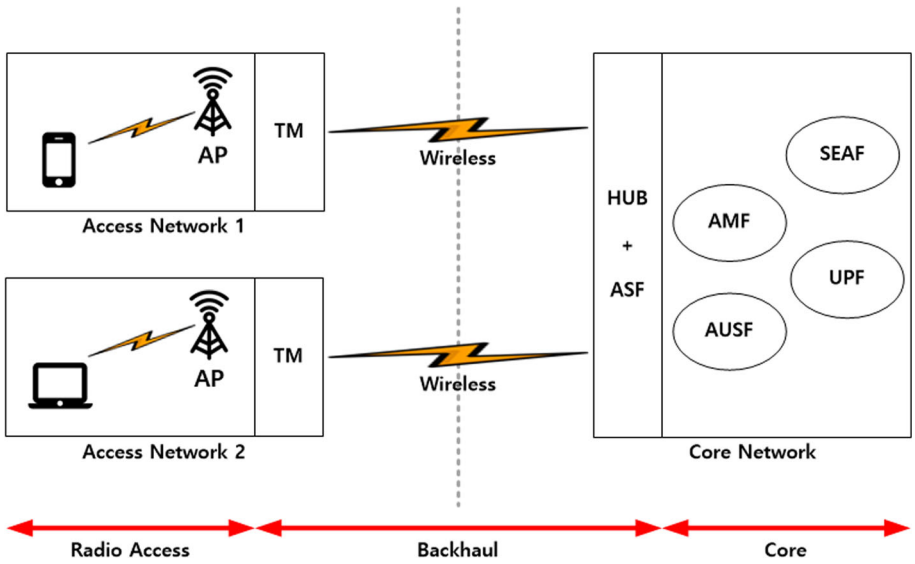

Fig. 1 5G wireless backhaul network architecture

secure key exchange, and perfect forward secrecy. More importantly, it is also designed to provide adaptive security by dynamically updating the security policy including cryptographic algorithms, authentication strength, key size, key lifetime, and so forth. For such goals, the proposed protocol consists of three phases: the initial authentication, key update, and policy update phases. In the first phase, a terminal TM and its HUB count on their ASF which serves as the trusted third party to mutually authenticate each other and exchange session keys. The second phase allows the two entities, TM and HUB, to efficiently update their session keys without the ASF's involvement while the third phase adjusts the security policy based on the security level so that the two entities can adaptively protect communication. It is worth to note that the ASF determines the security level in real time whenever necessary by evaluating the TM's profile and capability, the application data traffic's security requirements, and the current situation and authorization rule of the backhaul system and environments.

The proposed protocol has the following assumptions: (i) An ASF serves as an authentication server that both HUB and TM trust. For this goal, each TM shares its ID and long-term key with an ASF by registering itself into that function in advance. Also, it is assumed that each HUB establishes a secure channel with an ASF in advance. (iii) In order to use timestamp, the involved entities, TM, HUB, and ASF, are time-synchronized. The notations used in this paper are shown in Table 1.

Table 1 Notations table

Notation	Meaning
TM	Terminal
HUB	HUB
ASF	Authentication Server Function
IDX	X's identifier
Capability	List of the possible cryptographic algorithms, authentication and key exchange methods, etc.
Policy	The adaptively selected cryptographic algorithms, authentication and key exchange method, key size, key lifetime, etc. to secure the 5G backhaul system
PMK	Pre-master key
MK	Master key
AK	Authentication Key
CK	Cipher key
CM	AES128-CMAC
nx	xth nonce
ts	Time stamp
Kold	Secret key generated in previous session
$\|\|$	Concatenation
[]	Optional parameter
p, g	p is a pre-configured public modulus (prime) and g is a pre-configured public base for Diffie-Hellman key exchange
PFS	Perfect Forward Secrecy
$K_{X\text{-}Y}$	Shared secret key between X and Y.

3.2 Threat model

The threat model for the proposed protocol is based on the Dolev-Yao adversary model [29], which is one of the most well-known adversary models. Therefore, an adversary in the protocol can get all the messages transmitted over networks. To launch attacks, those messages can be modified, deleted, forged, replayed, injected, or redirected. However, the adversary is not allowed to decrypt and interpret an encrypted message if she or he fails to get the corresponding key.

3.3 Initial authentication phase

The initial authentication phase depicted in Fig. 2 aims to allow the TM and the HUB to mutually authenticate each other with the help of their ASF. Note that the ASF shares a long-term key $K_{\mathrm{TM-ASF}}$ with the TM in advance as well as pre-establishes a secure channel with the HUB. Especially, during this phase, the ASF dynamically decides the security level, based on which it adaptively makes the security policy. According to the decided security level, this phase can optionally deploy the Diffie-Hellman key exchange to establish a pre-master key PMK between the TM and the HUB in a way that a perfect forward secrecy is provided. In other words, the Diffie-Hellman key exchange is applied if recommended by the determined security level to strengthen the authentication and key exchange. In this case, the two parties remove their Diffie-Hellman private keys immediately after this phase so that the session keys cannot be recovered later.

- The TM starts this phase by sending the Auth_Req message to the HUB, which is then transmitted to the ASF. For this goal, the TM obtains the current time information ts, generates an arbitrary random number n1, and prepares for the capability attribute with its possible cryptographic algorithms and so forth.Once these values are initialized, the TM adds $\mathrm{ID_{TM}}$ and $\mathrm{ID_{HUB}}$ to them and calculates the message authentication code CM1 (by means of CMAC) with the long-term key $\mathrm{K_{TM-ASF}}$. The TM then sends the Auth_Req message to the HUB where the included CM1 is used primarily for the integrity and authenticity of the message to be sent to the ASF via the HUB. When the HUB receives the message, it checks if the IDHUB is correct and ts is within the allowed time window. Only if valid, it transfers the Auth_Req message to the ASF. In this way, the HUB can defend against the resource exhaustion attack, a kind of DoS attack, caused by excessive message flooding. Upon a receipt of the Auth_Req message, the ASF verifies that $\mathrm{ID_{HUB}}$ is the HUB's identifier and ts is valid. In turn, it verifies the CM1 through the shared secret key $\mathrm{K_{TM-ASF}}$. If the above verification is positive, the ASF can trust the TM, and based on the trust it proceeds the next step. In other words, the TM is now authenticated to the ASF.
- In order to prepare for the ASF_Auth_Res message, the ASF evaluates the TM's capability and profile, the application data traffic's security requirements, and the current situation and authorization rule of its backhaul system and environments, thereby deciding the security level. Then, based on the decided security level, it creates the policy by not only deciding cryptographic algorithms, key size, key lifetime, etc., but also choosing whether the Diffie-Hellman key exchange is used or not. Note that the security of the authentication and key exchange can be enhanced if the Diffie-Hellman key exchange is adopted. At the same time, it generates the pre-master key PMK by computing $\mathrm{CM}(\mathrm{K_{TM-ASF}}, \mathrm{ts}||\text{“pre-master key”})$. Afterwards, the ASF returns the policy and the PMK in the form of the ASF_Auth_Res message to the HUB over the secure channel.
- Upon receiving the ASF_Auth_Res message, the HUB first checks the included policy, from which it adopts the selected cryptographic algorithms, key size, key lifetime, etc. to protect the data communication after this phase. At this point, if recommended, it prepares for the Diffie-Hellman key exchange by choosing its private key y and computing the corresponding public key g^y. In addition, the HUB generates a random number n2 and uses the included PMK to calculate the master session key $\mathrm{MK} = \mathrm{CM}(\mathrm{PMK}, n1||n2||\text{“master key”})$. Then, with the session key, it computes the CMAC value $\mathrm{CM2} = \mathrm{CM}(\mathrm{MK}, \mathrm{ID_{HUB}}||\mathrm{ID_{TM}}||n1||n2||\mathrm{policy}[\,||g^y])$, followed by sending the Auth_Res message to the TM.
- When the TM receives the Auth_Res message, it checks if the included $n1$ matches with the initial random number sent by itself, and then computes the PMK and the MK as done by the ASF and the HUB, respectively. Afterwards, the accompanied CM2 is validated with the MK to verify if the Auth_Res message is fresh and authentic. Aside from trusting the message, the TM, if the above validation is positive, can successfully authenticate the HUB because it cannot generate the CM2 without receiving the PMK from the ASF. As the next step, the TM computes the authentication and cipher session keys (AK and CK, respectively) with the MK and the two nonces n1 and n2. At this point, if the given policy recommends the Diffie-Hellman key exchange, the TM should randomly generate its private key x and the corresponding public key g^x, and then involve g^{xy} for generating the two session keys AK and CK. Note that in this case, the mutual authentication and key exchange between the HUB

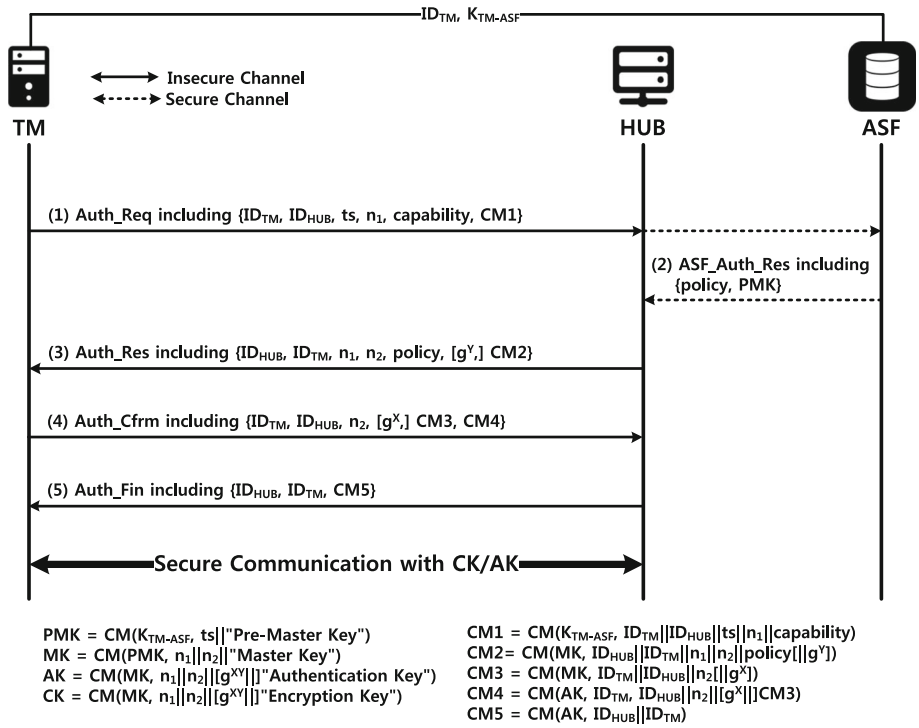

Fig. 2 Initial authentication phase

and the TM can be enhanced with the Diffie-Hellman key exchange. Moreover, the TM achieves the perfect forward secrecy by removing the private key x from its memory immediately after computing the two keys. On the other hand, only if the CM2 is valid, the TM performs the expensive Diffie-Hellman public key operations, which can prevent the resource exhaustion attacks, a kind of DoS attack. Finally, the TM sends the HUB the Auth_Cfrm message as a session key confirm message. This message, unlike other ones, includes the two CMAC values CM3 and CM4 to confirm that the TM owns the two keys MK and AK, respectively. In particular, if the Diffie-Hellman key exchange is used, the HUB can counter the resource exhaustion attacks by first validating the CM3 prior to the expensive public key operations.

- Once the Auth_Cfrm message arrives, the HUB first uses the MK to test if its CM3 is correct. If the test is successful, the HUB can authenticate the TM and trust its ownership for the MK. Then, the two session keys AK and CK are derived in the same way as done by the TM. The HUB in turn validates the involved CM4 to confirm that the TM owns the AK, which indicates also the TM's ownership for the CK. Here, if the Diffie-Hellman key exchange is requested, the HUB should compute g^{xy} and reflect it on the session keys AK and CK. In this case, the authentication and key exchange can be strengthened. If the CM4 is valid, the HUB concludes this phase by sending the Auth_Fin message as a key confirmation message. On arrival of this message, the TM validates its CM5 to confirm that both the AK and the CK are agreed well between the HUB and itself. At this point, the message cannot be replayed because the CM5, which is computed with the new session key AK, is fresh.

3.4 Key update phase

This phase, shown in Fig. 3, is triggered by the TM to update its authentication and cipher session keys AK and CK when their lifetime is expired. Note that without the ASF's involvement, this phase updates only the two session keys while keeping the current security policy. Similar to the initial authentication phase, if the Diffie-Hellman key exchange is applied, the TM and the HUB delete their private keys after negotiating the session keys CK and AK.

- The TM starts this phase by sending the KU_Req message. For this, it prepares for a random number n1 and timestamp ts while optionally generating its Diffie-Hellman public key pair (x, g^x) according to the policy set in the initial authentication phase or the policy update phase. Then, the current session key AK_{old} is used to compute the CMAC value CM1

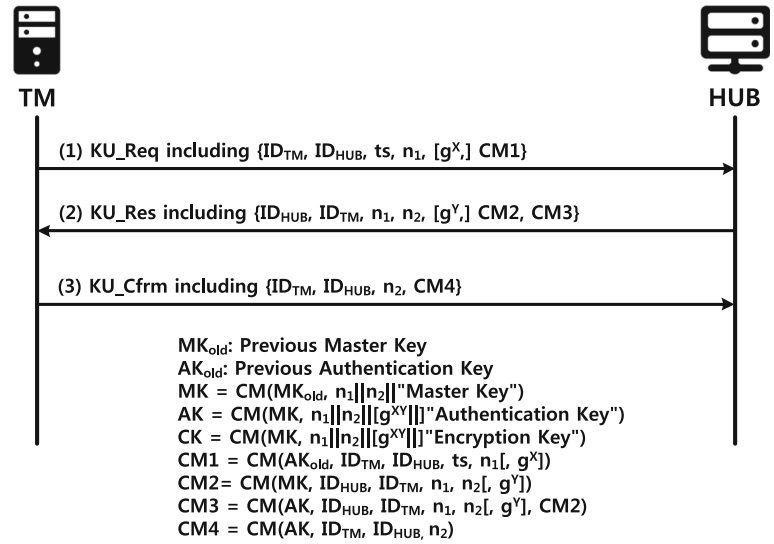

Fig. 3 Key update phase

on the above information and the IDs of the TM and the HUB. On receiving the KU_Req message, the HUB validates the CM1 to gain the belief that the TM really wants to update the session keys. Therefore, if the CM1 is valid, it computes the new master session key MK and the new two session keys AK and CK. Even though the Diffie-Hellman key exchange is requested, such session key generation is not vulnerable to the resource exhaustion attacks by performing the public key operations after validating the CM1. Once the new session keys are successfully generated, the HUB computes the two CMAC values CM2 and CM3 where the CM2 shows the HUB owns the new MK and the CM3 shows the HUB owns both the new AK and CK. Finally, it sends the KU_Res message to the TM.

- Upon receiving the KU_Res message, the TM first checks if the included n1 is correct. In positive case, it generates the new master session key MK, and then uses that new key to verify that the CM2 is valid. If this verification is true, the TM can trust that it shares the new MK with the HUB. Based on such trust, the TM derives the new authentication and cipher session keys AK and CK where the Diffie-Hellman key exchange can be applied if recommended. Here, note that the KU_Res message is not vulnerable to the resource exhaustion attack because the TM generates the new session keys only if the CM2 is correct. Once obtaining the new session keys AK and CK, the TM utilizes the new AK to validate the CM3. If this validation is true, it can obtain the belief that the HUB agrees both the new AK and CK with itself.
- The TM concludes this phase by sending the KU_Cfrm message whose CMAC value CM4 is computed with the new AK. When receiving the message, the HUB first verifies the included n2, then testing if the CM4 is correct. If positive, the HUB can confirm that the TM owns both the new session keys AK and CK as well as the key update process is successfully executed.

3.5 Policy update phase

This phase is initiated by a HUB whenever one of the associated TMs should update its policy due to change of its current security context or the system's security situation. As shown in Fig. 4, the HUB and the TM, based on the new policy, not only update the new cryptographic algorithms, key size, key lifetime, and etc., but also establish the new session keys MK, AK, and CK. In the case that the Diffie-Hellman key exchange is adopted, this phase makes sure that AK and CK cannot be re-calculated later by forcing the HUB and the TM to forget their private key as soon as the two keys are newly generated.

- If the ASF detects, after communicating with the HUB, the TM's current security context or the security status of the backhaul system and environments are remarkably changed, it decides the new security level by re-evaluating the TM's capability and profile, the current application data traffic's security requirements, the current situation and authorization rule of its backhaul system and environments, and so on. Then, the new policy, $\text{policy}_{\text{new}}$, is made according to the determined

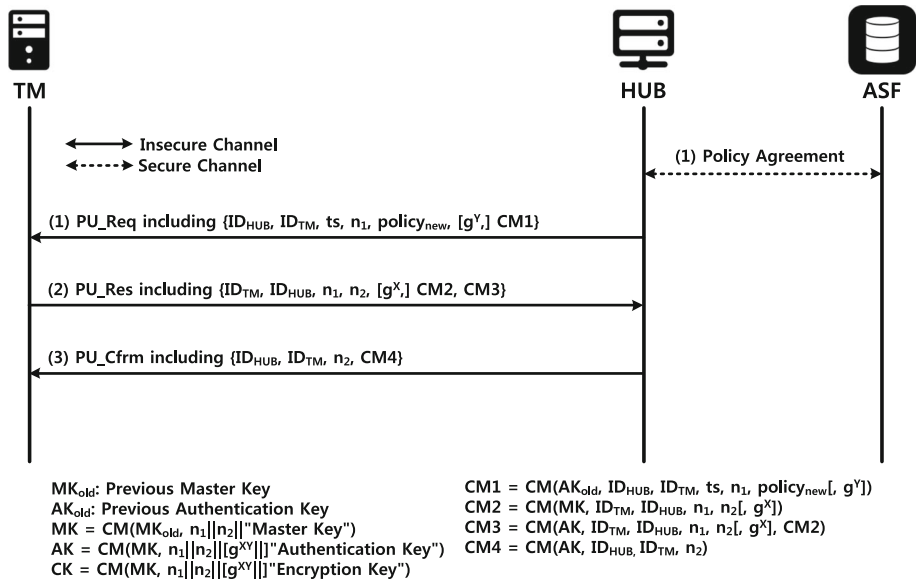

Fig. 4 Policy update phase

security level, and transferred to the HUB over the pre-established secure channel. The policy agreement indicates the above procedure. More details on this procedure is beyond this research.

- When the HUB receives the $\text{policy}_{\text{new}}$ from the ASF, it starts this phase by transmitting the PU_Req message, which is protected by the CM1. Moreover, this message recommends the TM to accept the $\text{policy}_{\text{new}}$. On arrival of this message, the TM validates the timestamp ts and the CM1. If successful, it authenticates the HUB and gains trust enough to adopt the $\text{policy}_{\text{new}}$ and compute the new session keys MK, CK, and AK. Once this key generation is completed, the new keys MK and AK are used to calculate the two CMAC values CM2 and CM3, respectively. It means that the CM2 indicates that the TM possesses the MK as well as the CM3 indicates that the TM possesses the AK and the CK. Moreover, if the Diffie-Hellman key exchange is applied, the HUB can prevent the resource exhaustion attacks by validating the CM2 prior to the needed public key operations. Finally, the TM sends the PU_Res message to the HUB.
- Upon a receipt of the PU_Res message, the HUB tests if the included n1 matches with what it originally sent. Once this test holds, the HUB first generates the new master session key MK, which is then used to validate the CM2. The CM2, if it is shown to be valid, ensures the HUB that the MK is shared well between the TM and itself. That makes the HUB generate two more new session keys AK and CK, then computing the CM3 with the AK. Here, if the Diffie-Hellan key exchange is applied, the CM2 enables the HUB to defend against the resource exhaustion attacks as mentioned above. Once conforming that the CM3 is valid, the HUB can conclude that both the AK and the CK are agreed between the TM and itself. As a result, based on the CM2 and CM3, the HUB can authenticate the TM and gain the trust on the new session keys. Moreover, it can confirm that the TM adopts the $\text{policy}_{\text{new}}$. As the next step, the HUB transmits the PU_Cfrm message to ensure the TM that it has the new authentication session key AK. At this point, the meaning that the HUB has the AK indicates that the HUB has also the MK and the CK.
- On receiving the PU_Cfrm message, the TM first checks if the included n2 and CM4 are valid. If this validation is successful, the authentication of the HUB to the TM, which is first performed based on AK_{old} in (2), is enhanced through the CM4 computed with the AK. More importantly, the TM can confirm that the new session keys are not only agreed between the HUB and itself, but also the HUB is going to reflect the $\text{policy}_{\text{new}}$ from now on.

4 Security analysis

In this section, the proposed protocol's correctness is thoroughly analyzed through the two formal security verification methods, BAN-logic [30, 31] and Scyther[32].

4.1 Security analysis based on BAN-logic

Since introduced by Burrows, Abadi, and Needham, BAN-logic has been widely applied to security analysis on many security protocols due to its simplicity, intuitive,

and robust [33–35]. For BAN-logic-based security analysis, a security protocol is (i) translated into an idealization form, (ii) its assumptions and goals are defined, and (iii) the BAN-logic rules are repeatedly applied to the idealized protocol to evolve its beliefs until the goals are satisfied. Tables 2 and 3 show the notations and inference rules of BAN-logic.

Here, we formally verify the three phases based on BAN-logic while assuming that the Diffie-Hellman key exchange is applied.

4.1.1 Initial authentication phase

As the first step, the initial authentication phase is idealized as follows.

(I1) $\langle \mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, \mathrm{ts}, n_1, \mathrm{capability} \rangle_{K_{\mathrm{TM-ASF}}}$

(I2) $\left\langle \mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_1, n_2, \mathrm{policy}, g^y, \mathrm{TM} \overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{MK}}$

(I3) $\left\langle \mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, n_2, g^x, \mathrm{TM} \overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{MK}}$,
$\left\langle n_2, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{AK}}$

(I4) $\left\langle D_{\mathrm{HUB}}, \mathrm{ID_{TM}}, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{AK}}$

The assumptions are defined as follows. Here, (AI3) and (AI7) are made because the ASF, if the TM is successfully authenticated, secure sends the HUB the pre-master key PMK, from which the HUB then derives the MK. Moreover, it is reasonable to add (AI10) because the AK is derived from n_1, which the TM believes to be fresh.

(AI1) ASF believes TM $\overset{K_{\mathrm{TM-ASF}}}{\Longleftrightarrow}$ ASF
(AI2) ASF believes #(ts)
(AI3) TM believes TM $\overset{\mathrm{MK}}{\Longleftrightarrow}$ HUB
(AI4) TM believes #(n_1)
(AI5) TM believes $\overset{g^x}{\rightarrow}$ TM
(AI6) HUB believes TM $\overset{\mathrm{MK}}{\Longleftrightarrow}$ HUB
(AI7) HUB believes #(n_2)
(AI8) HUB sees n_1

Table 2 Notations of BAN logic

Notation	Meaning
P **believes** *X*	*P* believes the message *X* and acts as if it is true
P **sees** *X*	*P* receives the message *X*
P **said** *X*	*P* previously sent the message *X*
P **controls** *X*	*P* has authority on *X*
#(X)	*X* is fresh
$P \overset{K}{\leftrightarrow} Q$	*K* is a secret key shared between *P* and *Q*
$\overset{K}{\rightarrow} P$	*K* is the *P*'s public key
$P \overset{K}{\Leftrightarrow} Q$	K is a shared secret between P and Q.
$\{X\}_K$	*X* is encrypted with *K*
$\langle X \rangle_K$	*X* is combined with a secret *K*

Table 3 Rules of BAN logic

Rule	Formula
MM: Message Meaning Rule	$\frac{P \text{ believes } P \overset{K}{\leftrightarrow} Q,\ P \text{ sees } \{X\}_K}{P \text{ believes } Q \text{ said } X}$ $\frac{P \text{ believes } P \overset{K}{\Leftrightarrow} Q,\ P \text{ sees } \langle X \rangle_K}{P \text{ believes } Q \text{ said } X}$ $\frac{P \text{ believes } \overset{K}{\rightarrow} Q,\ P \text{ sees } \{X\}_{Q^{-1}}}{P \text{ believes } Q \text{ said } X}$
NV: Nonce Verification Rule	$\frac{P \text{ believes } \#(X),\ P \text{ believes } Q \text{ said } X}{P \text{ believes } Q \text{ said } X}$
JR: Jurisdiction Rule	$\frac{P \text{ believes } Q \text{ controls } X,\ P \text{ believes } Q \text{ believes } X}{P \text{ believes } X}$
FR: Freshness Rule	$\frac{P \text{ believes } \#(X)}{P \text{ believes } \#(X,Y)}$
DR: Decomposition Rule	$\frac{P \text{ sees } (X,Y)}{P \text{ sees } X}$
BC: Belief Conjunction Rule	$\frac{P \text{ believes } X,\ P \text{ believes } Y}{P \text{ believes } (X,Y)}$ $\frac{P \text{ believes } Q,\ P \text{ believes } (X,Y)}{P \text{ believes } Q \text{ believes } X}$ $\frac{P \text{ believes } Q \text{ said } (X,Y)}{P \text{ believes } Q \text{ said} X}$
DH: Diffie-Hellman Rule	$\frac{P \text{ believes } Q \text{ said} \overset{g^Y}{\rightarrow} Q,\ P \text{ believes } \overset{g^X}{\rightarrow} P}{P \text{ believes } P \overset{g^{XY}}{\leftrightarrow} Q}$ $\frac{P \text{ believes } Q \text{ said} \overset{g^Y}{\rightarrow} Q,\ P \text{ believes } \overset{g^X}{\rightarrow} P}{P \text{ believes } P \overset{g^{XY}}{\Leftrightarrow} Q}$

(AI9) TM believes #(TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB)

We make the following goals where (G1) means the authentication of the TM to the ASF, (G2) and (G4) describe the mutual authentication between the TM and the HUB, (G3) indicates the TM's verification of the policy, and (G5)–(G12) express the exchanges of the session keys AK and CK.

(G1) ASF believes TM believes [$\mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, \mathrm{ts}, n_1,$ capability]

(G2) TM believes HUB believes[$\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, \mathrm{n_1}, \mathrm{n_2},$ policy, g^Y, TM $\overset{\mathrm{MK}}{\Longleftrightarrow}$ HUB]

(G3) TM believes HUB believes policy

(G4) HUB believes TM believes [$\mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, n_2, g^x,$ TM $\overset{\mathrm{MK}}{\Longleftrightarrow}$ HUB]

(G5) TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB
(G6) TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB
(G7) HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB
(G8) HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB
(G9) HUB believes TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB
(G10) HUB believes TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB
(G11) TM believes HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB
(G12) TM believes HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB

From (I1), we derive

(D1) ASF sees $\langle \mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, \mathrm{ts}, n_1, \mathrm{capability} \rangle_{K_{\mathrm{TM-ASF}}}$

(D2) ASF believes TM believes [$\mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, \mathrm{ts}, n_1,$ capability]

by (D1),(AI1), MM, (AI2), FR, NV

From (I2), we derive

(D3) TM sees $\langle \mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_1, n_2, \text{policy}, g^y, \mathrm{TM} \overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUB} \rangle_{\mathrm{MK}}$

(D4) TM believes HUB said $\left[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_1, n_2, \text{policy}, g^y, TM \overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUB}\right]$

by (D3),(AI3), MM

(D5) TM believes HUB believes $[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_1, n_2, \text{policy}, g^y, \mathrm{TM} \overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUB}]$

(D6) TM believes HUB believes policy by (D5), BC

(D7) TM believes HUB believes TM $\overset{\mathrm{MK}}{\Longleftrightarrow}$ HUB by (D5), BC

(D8) TMbelieves TM $\overset{g^{XY}}{\Longleftrightarrow}$ HUB by (D4), BC, (AI5),DH

(D9) TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB by (D8), (D5), BC, (AI4)

(D10) TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB by (D8), (D5), BC, (AI4)

From (I3), we derive

(D11) HUB sees $\left[\left\langle \mathrm{ID}, \mathrm{ID_{HUB}}, n_2, g^x, \mathrm{TM} \overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{MK}'} \left\langle n_2, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{AK}}\right]$

(D12) HUB believes TM said $\left[\overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUBID_{TM}}, \mathrm{ID_{HUB}}, n_2, g^x, \mathrm{TM} \overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUB}\right]$

by (D11), (AI6), MM

(D13) HUB believes TM believes $\left[ID_{\mathrm{TM}}, \mathrm{ID_{HUB}}, n_2, g^x, \mathrm{TM} \overset{\mathrm{MK}}{\Longleftrightarrow} \mathrm{HUB}\right]$

by (D12), (AI7), FR, NV

(D14) HUB believes TM believes TM $\overset{\mathrm{MK}}{\Longleftrightarrow}$ HUB by (D13), BC

(D15) HUB believes TM $\overset{g^{XY}}{\Longleftrightarrow}$ HUB by (D12), BC, (AI8), DH

(D16) HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB by (D15), (AI7), (AI8)

(D17) HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB by (D15), (AI7), (AI8)

(D18) HUB believes TM believes $\left[n_2, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB}\right]$

by (D11), (D16), MM, (AI7), FR, NV

(D19) HUB believes TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB by (D18), BC

(D20) HUB believes TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB by (D18), BC

From (I4), we derive

(D21) TM sees $\left\langle D_{\mathrm{HUB}}, \mathrm{ID_{TM}}, \mathrm{TM} \overset{AK}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB}\right\rangle_{AK}$

(D22) TM believes HUB believes $\left[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB}\right]$

by (D21), (D9), MM, (AI10), FR, NV

(D23) TM believes HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB by (D22), BC

(D24) TM believes HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB by (D22), BC

From the obtained beliefs (D1)∼(D24), we can see that the initial authentication phase can satisfy the goals defined above. In addition, the following lemmas can be derived, thereby showing that this phase achieves the security requirements mentioned above.

Lemma 1 *The initial authentication phase provides mutual authentication between the TM and the HUB.*

Proof (D5) and (D22) show that the HUB is authenticated to the TM based on the session key MK and AK, respectively. Similarly, (D13) shows that the TM is authenticated to the HUB based on AK. Hence, we can conclude that the initial authentication phase achieves the mutual authentication between two entities. □

Lemma 2 *The initial authentication phase enables the TM and the HUB to successfully negotiate the session keys AK and CK.*

Proof The two parties, TM and HUB, obtain the beliefs (D9), (D10), (D16), and (D17) that indicate that the session keys CK and AK are well exchanged. They are completed through the beliefs (D19), (D20), (D23), and (D24) that mean that each party trusts the correspondent's belief on the keys. Hence, it can be seen that this lemma holds. □

Lemma 3 *The initial authentication phase provides the perfect forward secrecy for the session keys AK and CK.*

Proof (D8) and (D15) show that g^{XY} is successfully exchanged based on the Diffie-Hellman exchange. As mentioned above, the two parties' private key is forgotten after this phase. Hence, we can say that the perfect forward secrecy for g^{XY} can be satisfied. That leads based on (D9), (D10), (D16), and (D17) to the conclusion that Lemma 3 holds. □

Lemma 4 *The initial authentication phase defends against resource exhaustion attacks.*

Proof In this phase, the resource exhaustion attacks can happen when the TM or the HUB should perform the

Diffie-Hellman key exchange. Assume that the Diffie-Hellman key exchange is adopted in this phase. Before sending the Auth_Res message, the HUB should generate its private key Y. At this point, the HUB can count on the ASF_Auth_Res message sent from the ASF to avoid these attacks. Therefore, to take the resource exhaustion attack into consideration, we can focus on the Auth_Res and Auth_Cfrm messages, which are protected with the CMAC values CM2 and CM3 computed with MK. Only if the values are valid, the involved parties perform the associated expensive operations while preventing the resource exhaustion attacks. The derived beliefs (D5) and (D13) show that CM2 and CM3 are successfully verified respectively, thereby leading to the conclusion that Lemma 4 holds. □

Lemma 5 *The initial authentication phase provides confidentiality and integrity.*

Proof In this phase, confidentiality indicates that the session keys CK and AK are securely exchanged without any leakage. From Lemma 2, we can see that the two keys are successfully exchanged, which can be enhanced through Lemma 3 in a way that the perfect forward secrecy is guaranteed if the Diffie-Hellman key exchange is adopted. On the other hand, integrity means that the CMAC values CM1–CM5 are valid. It can be shown from the obtained beliefs (D2), (D5), (D13), (D18), and (D22) that all the CMAC values are correct, which thus integrity can be supported. As a result, it can be proved that Lemma 5 holds. □

4.1.2 Key update phase

This phase is translated into an idealized form while its assumptions are defined as shown below.

(K1) $\langle \mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, \mathrm{ts}, n_1, g^x \rangle_{\mathrm{AK_{old}}}$

(K2) $\langle \mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_1, n_2, g^y \rangle_{\mathrm{MK}}, \langle n_1, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \rangle_{\mathrm{AK}}$

(K3) $\langle \mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, n_2, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \rangle_{\mathrm{AK}}$

(AK1) HUB believes TM $\overset{\mathrm{AK_{old}}}{\Longleftrightarrow}$ HUB

(AK2) HUB believes #(ts)

(AK3) HUB believes $\overset{g^Y}{\rightarrow}$ HUB

(AK4) TM believes TM $\overset{\mathrm{MK}}{\Longleftrightarrow}$ HUB

(AK5) TM believes #(n_1)

(AK6) HUB believes $\overset{g^X}{\rightarrow}$ TM

(AK7) HUB believes #(n_2)

(AK8) HUB believes TM $\overset{\mathrm{MK}}{\Longleftrightarrow}$ HUB

(AK4) and (AK9) are added because the TM and the HUB know the previous master key MKold, thereby being able to derive the new one MK. The goals for this phase are defined as follows. (G13) and (G14) indicate the mutual authentication between the HUB and the TM while – (G19) mean the session key exchange for the two session keys, CK and AK.

(G13) HUB believes TM believes $\left[\overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TMID_{HUB}}, \mathrm{ID_{TM}}, n_2, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \right]$

(G14) TM believes HUB believes $\left[n_1, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \right]$

(G15) HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB

(G16) HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB

(G17) TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB

(G18) TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB

(G19) HUB believes TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB

(G20) HUB believes TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB

(G21) TM believes HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB

(G22) TM believes HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB

From (K1), we derive

(D25) HUB sees $\langle \mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, ts, n_1, g^x \rangle_{\mathrm{AK_{old}}}$

(D26) HUB believes TM said $\left[\mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, ts, n_1, g^x \right]$ by (D25), (AK1), MM

(D27) HUB believes TM believes $\left[\mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, \mathrm{ts}, n_1, g^x \right]$ by (D26), (AK2), FR, NV

(D28) HUB believes TM $\overset{g^{XY}}{\Longleftrightarrow}$ HUB (D26), BC, (AK3), DH

(D29) HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB (D28), (D27), BC, (AK7)

(D30) HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB (D28), (D27), BC, (AK7)

From (K2), we derive

(D31) TM sees $\left[\langle ID_{\mathrm{HUB}}, \mathrm{ID_{TM}}, n_1, n_2, g^Y \rangle_{\mathrm{MK}}, \left\langle n_1, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_A K \right]$

(D32) TM believes HUB said $\left[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_1, n_2, g^Y \right]$ by (D31), (AK4), MM

(D33) TM believes HUB believes $\left[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_1, n_2, g^Y \right]$ by (D32), (AK5), FR, NV

(D34) TM believes TM $\overset{g^{XY}}{\Longleftrightarrow}$ HUB by (D32), BC, (AK6), DH

(D35) TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB by (D34), (D33), BC, (AK5)

(D36) TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB by (D34), (D33), BC, (AK5)

(D37) TM believes HUB believes $\left[n_1, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \right]$ by

(D31), (D35), MM, (AK5), FR, NV

(D38) TM believes HUB believes TM $\overset{AK}{\Longleftrightarrow}$ HUB by (D37), BC

(D39) TM believes HUB believes TM $\overset{CK}{\Longleftrightarrow}$ HUB by (D37), BC

From (K3), we derive

(D40) HUB sees $\langle \mathrm{ID}, \mathrm{ID}_{\mathrm{HUB}}, n_2, \mathrm{TM} \overset{AK}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{CK}{\Longleftrightarrow} \mathrm{HUB} \rangle_{\mathrm{AK}}$

(D41) HUB believes TM believes $\left[\overset{AK}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TMID}_{\mathrm{HUB}}, \mathrm{ID}_{\mathrm{TM}}, n_2, \mathrm{TM} \overset{AK}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{CK}{\Longleftrightarrow} \mathrm{HUB} \right]$ by (D40), (D29), MM, (AK7), FR, NV

(D43) HUB believes TM believes TM $\overset{AK}{\Longleftrightarrow}$ HUB by (D42), BC

(D44) HUB believes TM believes TM $\overset{CK}{\Longleftrightarrow}$ HUB by (D42), BC

It is proved from the above derived beliefs (D25)–(D44) that the key update phase satisfies the goals (G13)–(G22). Based on the above verification, the below lemmas can be obtained.

Lemma 6 *The key update phase provides mutual authentication between the TM and the HUB.*

Proof (D27) and (D33) demonstrate that the mutual authentication between the HUB and the TM is successfully performed based on $\mathrm{AK}_{\mathrm{old}}$ and MK. Furthermore, (D37) and (D41) show that the two parties successfully authenticate each other based on AK. The latter enhances the former strongly enough to lead to the conclusion that this lemma holds. □

Lemma 7 *The key update phase enables the TM and the HUB to successfully negotiate the session keys AK and CK.*

Proof The beliefs (D29), (D30), (D35), and (D36) show that the session keys CK and AK are well agreed between the two parties TM and HUB. They are further evolved and finalized through the beliefs (D38), (D39), (D43), and (D44) indicating that the parties believe each other's belief on the keys. As a result, we can conclude that this lemma holds. □

Lemma 8 *The key update phase provides the perfect forward secrecy for the session keys AK and CK.*

Proof From (D28) and (D34), it can be seen that g^{XY} is successfully negotiated based on the Diffie-Hellman exchange. In addition, after this key exchange, the ephemeral private keys X and Y are immediately deleted from the two parties' memory. Accordingly, we can say that the perfect forward secrecy for g^{XY} can be fulfilled. That can be applied with the additional beliefs (D29), (D30), (D35), and (D36) to conclude that Lemma 8 holds. □

Lemma 9 *The key update phase defends against resource exhaustion attacks.*

Proof This phase is designed to prevent the resource exhaustion attacks when the Diffie-Hellman key exchange is used by allowing the key exchange only if the corresponding CMAC values, CM1 and CM2, are valid. The beliefs (D27) and (D33) prove that CM1 and CM2 are successfully verified respectively, thereby leading to the conclusion that Lemma 9 holds. □

Lemma 10 *The key update phase provides confidentiality and integrity.*

Proof In key update phase, confidentiality is regarded as securely negotiating the session keys CK and AK without any leakage. In such a context, we can see based on Lemmas 8 and 9 that this phase provides confidentiality. On the other hand, integrity is regarded as ensuring that all the CMAC values CM1–CM4 are valid. It can be shown from the obtained beliefs (D27), (D33), (D37), and (D41) that all the CMAC values are valid. As a result, it can be proved that Lemma 10 holds. □

Policy update phase: At first, we idealized this phase into the following form

(P1) $\langle \mathrm{ID}_{\mathrm{HUB}}, \mathrm{ID}_{\mathrm{TM}}, \mathrm{ts}, n_1, \mathrm{policy}_{\mathrm{new}}, g^Y \rangle_{\mathrm{AK}_{\mathrm{old}}}$

(P2) $\langle \mathrm{ID}, \mathrm{ID}_{\mathrm{HUB}}, n^1, n^2, g^X \rangle_{\mathrm{MK}}, \left\langle n^1, \mathrm{TM} \overset{CK}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{AK}}$

(P3) $\left\langle \mathrm{ID}_{\mathrm{HUB}}, \mathrm{ID}_{\mathrm{TM}}, n_2, \mathrm{TM} \overset{AK}{\Longleftrightarrow} \mathrm{HUB} \overset{CK}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{AK}}$

Then, we make the assumptions as shown below. Here (AP4) and (AP8) are added because the TM and the HUB can derive the new one MK from the previous master key $\mathrm{MK}_{\mathrm{old}}$.

(AP1) TM believes TM $\overset{\mathrm{AK}_{\mathrm{old}}}{\Longleftrightarrow}$ HUB

(AP2) TM believes #(ts)

(AP3) TM believes $\overset{g^X}{\rightarrow}$ TM

(AP4) TM believes $\#(n_2)$

(AP5) HUB believes TM $\overset{MK}{\Longleftrightarrow}$ HUB

(AP6) HUB believes $\#(n_1)$

(AP7) HUB believes $\overset{g^Y}{\rightarrow}$ TM

(AP5) TM believes TM $\overset{MK}{\Longleftrightarrow}$ HUB

```
role TM{

    fresh x, ts, n1: Nonce;
    var n2: Nonce;
    var Gy: Ticket;

    send_1(TM, HUB, TM, HUB, ts, n1, cm(Kta, TM, HUB, ts, n1));
    recv_4(HUB, TM, HUB, TM, n1, n2, Gy, cm(MK, HUB, TM, n1, n2, Gy));
    claim(TM, Running, HUB, cm(MK, HUB, TM, n1, n2, Gy), cm(cm(MK,n1,n2,h(Gy,x)),TM,HUB,n2,g(x),cm(MK,TM,HUB,n2,g(x))));
    send_15(TM, HUB, TM, HUB, n2, g(x), cm(MK, TM, HUB, n2, g(x)), cm(cm(MK,n1,n2,h(Gy,x)),TM,HUB,n2,g(x),cm(MK,TM,HUB,n2,g(x))));
    recv_16(HUB, TM, HUB, TM, cm(cm(cm(PMK, n1, n2), n1, n2, h(Gy, x)), HUB, TM));


    claim(TM, Alive);
    claim(TM, Nisynch);
    claim(TM, Niagree);
    claim(TM, Weakagree);
    claim(TM, Commit, HUB, cm(MK, HUB, TM, n1, n2, Gy), cm(cm(MK,n1,n2,h(Gy,x)),TM,HUB,n2,g(x),cm(MK,TM,HUB,n2,g(x))));
    claim(TM, Secret, Kta);
    claim(TM, Secret, PMK);
    claim(TM, Secret, MK);
    claim(TM, SKR, cm(MK, n1, n2, h(Gy,x)));

}
```

Fig. 5 Policy update phase

The goals for this phase are made as follows:

(G23) TM believes HUB believes $\left[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_2, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB}\right]$

(G24) HUB believes TM believes $\left[n_1, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{CK}{\Longleftrightarrow} \mathrm{HUB}\right]$

(G25) TM believes HUB believes$\mathrm{policy_{new}}$

(G26) HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB

(G27) HUB believes TM $\overset{CK}{\Longleftrightarrow}$ HUB

(G28) TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB

(G29) TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB

(G30) HUB believes TM believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB

(G31) HUB believes TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB

(G32) TM believes HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB

(G33) TM believes HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB

Here, (G23) and (G24) express the mutual authentication between the TM and the HUB, (G25) indicates that the new policy $\mathrm{policy_{new}}$ is verified by the TM, and (G26)-(G33) mean that the session keys AK and CK are securely negotiated. From (P1), we derive

(D45) TM sees $\langle \mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, \mathrm{ts}, n_1, \mathrm{policy_{new}}, g^Y \rangle_{\mathrm{AK_{old}}}$

Table 4 Claim event description

Event	Security property
Alive	Authentication
Nisynch	Authentication
Niagree	Authentication
Weakagree	Authentication
Running / commit	Authentication
Secret	Secrey
SKR	Secrey

(D46) TM believes HUB said $\left[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, \mathrm{ts}, n_1, \mathrm{policy_{new}}, g^Y\right]$ by (D45), (AP1), MM

(D47) TM believes HUB believes $\left[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, \mathrm{ts}, n_1, \mathrm{policy_{new}}, g^Y\right]$ by (D46), (AP2), FR, NV

(D48) TM believes HUB believes $\mathrm{policy_{new}}$ by (D47), BC

(D49) TM believes TM $\overset{g^{XY}}{\Longleftrightarrow}$ HUB by (D46), BC, (AP3), DH

(D50) TM believes TM} $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB by (D49), (D47), BC, (AP4)

(D51) TM believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB by (D49), (D47), BC, (AP4)

From (P2), we derive

(D52) HUB sees $\left[\langle \mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, n_1, n_2, g^X \rangle_{\mathrm{MK}} \left\langle n_1, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{AK}}\right]$

(D53) HUB believes TM said $\left[\mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, n_1, n_2, g^X\right]$ by (D52), (AP5), MM

(D54) HUB believes TM believes $\left[\mathrm{ID_{TM}}, \mathrm{ID_{HUB}}, n_1, n_2, g^X\right]$ by (D53), (AP6), FR, NV

(D55) HUB believes TM $\overset{g^{XY}}{\Longleftrightarrow}$ HUB by (D53), BC, (AP7), DH

(D56) HUB believes TM $\overset{\mathrm{AK}}{\Longleftrightarrow}$ HUB by (D55), (AP6), (D54), BC

(D57) HUB believes TM $\overset{\mathrm{CK}}{\Longleftrightarrow}$ HUB by (D55), (AP6), (D54), BC

(D58) HUB believes TM said $\left[n_1, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB}\right]$, by (D52), (D56), MM

(D59) HUB believes TM believes $\left[n_1, \mathrm{TM} \overset{\mathrm{AK}}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{\mathrm{CK}}{\Longleftrightarrow} \mathrm{HUB}\right]$, by (D58), (AP6), FR, NV

Claim				Status
p2mp	TM	p2mp,TM2	Alive	Ok
		p2mp,TM3	Weakagree	Ok
		p2mp,TM4	Commit HUB,Gy,g(x),n1,n2	Ok
		p2mp,TM5	Secret k(TM,ASF)	Ok
		p2mp,TM6	Secret cm(k(TM,ASF),ts)	Ok
		p2mp,TM7	Secret cm(cm(k(TM,ASF),ts),n1,n2,h(Gy,x))	Ok
		p2mp,TM8	SKR cm(cm(cm(k(TM,ASF),ts),n1,n2,h(Gy,x)),n1,n2)	Ok
	HUB	p2mp,HUB2	Alive	Ok
		p2mp,HUB3	Weakagree	Ok
		p2mp,HUB4	Commit TM,Gx,g(y),n1,n2	Ok
		p2mp,HUB5	Secret cm(k(TM,ASF),ts)	Ok
		p2mp,HUB6	Secret cm(cm(k(TM,ASF),ts),n1,n2,h(Gx,y))	Ok
		p2mp,HUB7	SKR cm(cm(cm(k(TM,ASF),ts),n1,n2,h(Gx,y)),n1,n2)	Ok
	ASF	p2mp,ASF1	Secret k(TM,ASF)	Ok
		p2mp,ASF2	Secret cm(k(TM,ASF),ts)	Ok

Fig. 6 Verification result of the initial phase

(D60) HUB believes TM believes TM $\overset{AK}{\Longleftrightarrow}$ HUB by (D59), BC
(D61) HUB believes TM believes TM $\overset{CK}{\Longleftrightarrow}$ HUB by (D59), BC

From (K3), we derive

(D62) TM sees $\left\langle \mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_2, \mathrm{TM} \overset{AK}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{CK}{\Longleftrightarrow} \mathrm{HUB} \right\rangle_{\mathrm{AK}}$

(D63) TM believes HUB believes $\left[\mathrm{ID_{HUB}}, \mathrm{ID_{TM}}, n_2, \mathrm{TM} \overset{AK}{\Longleftrightarrow} \mathrm{HUB}, \mathrm{TM} \overset{CK}{\Longleftrightarrow} \mathrm{HUB} \right]$

by (D62), (D50), MM, (AP4), FR, NV
(D64) TM believes HUB believes TM $\overset{AK}{\Longleftrightarrow}$ HUB by (D63), BC
(D65) TM believes HUB believes TM $\overset{CK}{\Longleftrightarrow}$ HUB by (D63), BC

According to the above verification, we can see that the policy update phase accomplishes the goals—(G22). Moreover, we can derive the below lemmas.

Lemma 11 *The policy update phase provides mutual authentication between the TM and the HUB.*

Proof It is seen from (D47) and (D54) that the mutual authentication between the HUB and the TM is successfully done based on AKold and MK. In addition, (D59) and (D63) indicate that the two parties successfully authenticate each other based on AK. The latter evolves the former strongly enough to lead to the conclusion that this lemma holds. □

Lemma 12 *The policy update phase enables the TM and the HUB to successfully negotiate the session keys AK and CK.*

Proof The beliefs (D49), (D50), (D56), and (D57) prove that the session keys CK and AK are well negotiated between the TM and the HUB. These beliefs are further enhanced and completed through the beliefs (D60), (D61), (D64), and (D65) indicating that the two parties believe

Claim				Status
p2mpkeyupdate	TM	p2mpkeyupdate,TM2	Alive	Ok
		p2mpkeyupdate,TM3	Nisynch	Ok
		p2mpkeyupdate,TM4	Niagree	Ok
		p2mpkeyupdate,TM5	Weakagree	Ok
		p2mpkeyupdate,TM6	Commit HUB,cm(cm(MKold,n1,n2),n1,n2,h(Gy,x))	Ok
		p2mpkeyupdate,TM7	Secret MKold	Ok
		p2mpkeyupdate,TM8	Secret AKold	Ok
		p2mpkeyupdate,TM9	SKR cm(MKold,n1,n2)	Ok
		p2mpkeyupdate,TM10	SKR cm(cm(MKold,n1,n2),n1,n2,h(Gy,x))	Ok
	HUB	p2mpkeyupdate,HUB2	Alive	Ok
		p2mpkeyupdate,HUB3	Nisynch	Ok
		p2mpkeyupdate,HUB4	Niagree	Ok
		p2mpkeyupdate,HUB5	Weakagree	Ok
		p2mpkeyupdate,HUB6	Commit TM,cm(cm(MKold,n1,n2),n1,n2,h(Gx,y))	Ok
		p2mpkeyupdate,HUB7	Secret MKold	Ok
		p2mpkeyupdate,HUB8	Secret AKold	Ok
		p2mpkeyupdate,HUB9	SKR cm(MKold,n1,n2)	Ok
		p2mpkeyupdate,HUB10	SKR cm(cm(MKold,n1,n2),n1,n2,h(Gx,y))	Ok

Fig. 7 Verification result of the key update phase

each other's belief on the session keys. Consequently, it can be shown that this lemma holds. □

Lemma 13 *The policy update phase provides the perfect forward secrecy for the session keys AK and CK.*

Proof From (D49) and (D55), it is proven that g^{XY} is successfully built based on the Diffie-Hellman exchange. Moreover, the ephemeral private keys X and Y are forgotten once the relevant session keys are agreed. Therefore, it can be seen that the perfect forward secrecy for g^{XY} is achieved. Together with this, (D29), (D30), (D35), and (D36) lead to the conclusion that Lemma 13 is accomplished. □

Lemma 14 *The policy update phase defends against resource exhaustion attacks.*

Proof When the Diffie-Hellman key exchange is applied, this phase first verifies the CMAC values CM1 and CM2 prior to the public key operations to prevent the resource exhaustion attacks. The beliefs (D47) and (D54) demonstrate that these CMAC values are successfully verified respectively. Hence, we can conclude that Lemma 14 holds. □

Lemma 15 *The policy update phase provides confidentiality and integrity.*

Proof In this phase, we can say that confidentiality is provided if the session keys CK and AK are securely exchanged without any leakage. Therefore, it is shown from Lemmas 12 and 13 that this phase provides confidentiality. On the other hand, here integrity means that all the CMAC values CM1–CM4 are valid, which can

Claim				Status
p2mppolicyupdate	TM	p2mppolicyupdate,TM2	Alive	Ok
		p2mppolicyupdate,TM3	Nisynch	Ok
		p2mppolicyupdate,TM4	Niagree	Ok
		p2mppolicyupdate,TM5	Weakagree	Ok
		p2mppolicyupdate,TM6	Commit HUB,cm(cm(MKold,n1,n2),n1,n2,h(Gy,x))	Ok
		p2mppolicyupdate,TM7	Secret MKold	Ok
		p2mppolicyupdate,TM8	Secret AKold	Ok
		p2mppolicyupdate,TM9	SKR cm(MKold,n1,n2)	Ok
		p2mppolicyupdate,TM10	SKR cm(cm(MKold,n1,n2),n1,n2,h(Gy,x))	Ok
	HUB	p2mppolicyupdate,HUB2	Alive	Ok
		p2mppolicyupdate,HUB3	Nisynch	Ok
		p2mppolicyupdate,HUB4	Niagree	Ok
		p2mppolicyupdate,HUB5	Weakagree	Ok
		p2mppolicyupdate,HUB6	Commit TM,cm(cm(MKold,n1,n2),n1,n2,h(Gx,y))	Ok
		p2mppolicyupdate,HUB7	Secret MKold	Ok
		p2mppolicyupdate,HUB8	Secret AKold	Ok
		p2mppolicyupdate,HUB9	SKR cm(MKold,n1,n2)	Ok
		p2mppolicyupdate,HUB10	SKR cm(cm(MKold,n1,n2),n1,n2,h(Gx,y))	Ok

Fig. 8 Verification result of the policy update phase

be proved by the obtained beliefs (D47), (D54), (D59), and (D63). As a result, we can conclude that Lemma 15 holds. □

So far the three phases, i.e., the proposed protocol, have been formally verified based on based on the BAN-logic. From the above verification, it is proven that the three phases are correct as well as satisfy the defined goals. Moreover, the derived 15 lemmas show that the proposed protocol fulfils the security requirements.

4.2 Security analysis based on Scyther

In the previous section, we formally validated the proposed protocol using BAN Logic and concluded that the protocol is secured against known attacks. However, formal methods, such as BAN Logic, that are used to verify security protocols have limitations in pointing out some essential protocol flaws[36, 37]. Hence, analyzing security protocols with only these formal methods cannot guarantee their trustworthiness.

Table 5 Comparison of protocols by security property

Security property	EAP-AKA	EAP-TLS	EAP-IKEv2	HIP	P2MP
SP1	O	O	O	O	O
SP2	O	O	O	O	O
SP3	O	O	O	O	O
SP4	O	O	O	O	O
SP5	X	X	O	O	O
SP6	X	O	O	X	O
SP7	X	X	X	X	O
SP8	O	X	X	X	O
Document	RFC 4187	RFC 5216	RFC 5106	RFC 5201	-

SP1 Confidentiality, *SP2* Integrity, *SP3* Mutual authentication, *SP4* Key exchange, *SP5* Perfect forward secrecy, *SP6* Key update; *SP7* Policy update, *SP8* Defense against resource exhaustion attack, *O* Support, *X* Not support

Consequently, other automated tools, such as Scyther, are needed to be applied to verify the desired security requirements of the proposed protocol.

Scyther is an automated tool for formal verification of security protocols. The target security protocol is modeled using an Security Protocol Description Language (SPDL) language and verified with an easy to use graphical user interface. When proving the possible claims of the protocol, the tool is capable of generating attack graphs, given attacks are found.

The target protocol expressed via SPDL is shown in Fig. 5. It consists of TM, HUB, and ASF entities. Each entity communicates with other entities through send and recv, verifies the behavior of the entity through claims, and demonstrates how secure the protocol is.

Claim events used in this paper can be categorized by functions as shown in Table 4, and the details are described in[38]. The claim events support verification of authentication and secrecy and can prove that the entity of the proposed protocol operates normally. In addition, it is possible to verify that keys used in the protocol are derived and exchanged securely.

The result of verifying the claim event of the protocol proposed in this paper is shown in Fig. 6. According to this, the proposed security protocol is safe from attack.

Figures 7 and 8 show the verification results of the key update phase and policy update phase of the proposed protocol, respectively. Source code for initial phase, key update phase, and policy update phase are included in Additional file 1: Appendix.

5 Performance analysis

Based on eight security properties, a comparative analysis of the proposed protocol with other four security protocols (EAP-TLS[39], EAP-AKA[23], EAP-IKEv2[40], and HIP[41] is shown in Table 5.

Table 7 Comparison of protocols by round trip time

Protocols	Communication overhead			
	TM		HUB	
	T1	T2	T3	T4
EAP-AKA	4.5RTT	4.5RTT	4RTT	4TT
EAP-TLS	8.5RTT	8.5RTT	8RTT	8RTT
EAP-IKEv2	6.5RTT	6.5RTT	6RTT	6RTT
HIP	3.5RTT	3.5RTT	3RTT	3RTT
P2MP	3.5RTT	3.5RTT	4RTT	4RTT

RTT round trip time between TM and HUB, ***T1*** time taken before TM can send data, *T2* time taken before TM can receive data, *T3* time taken before HUB can send data, *T4* time taken before HUB can receive data

The result of the analysis shows that even if all of the protocols compared with the proposed protocol support confidentiality, integrity, mutual authentication, and key exchange, they all failed to support policy update property. Moreover, perfect forward secrecy is only supported by EAP-IKEv2 and HIP, while EAP-AKA and HIP are failed to support key update. With regard to attack resistance, only EAP-AKA and HIP are resistant to resource exhaustion attack and malicious TM attacks, respectively. Accordingly, it can be concluded that the proposed protocol offers a better security by fulfilling all the nine security requirements.

Table 6 shows the computation overhead comparison of the proposed protocol against other security protocols. From the total computation overhead, it is observed that the computation cost of the proposed protocol is better than other public key-based schemes, EAP-TLS, and EAP-IKEv2. Even though EAP-AKA has lower computation overhead than others, it fails to fulfill important security requirements such as perfect forward secrecy, key, and policy update, and no malicious TM attack resistance.

Table 6 Comparison of protocols by computation overhead

Protocols	Computation overhead			
	TM	HUB	ASF	Total
EAP-AKA	$9C_{HM}$	–	$9C_{HM}$	$18C_{HM}$
EAP-TLS	$1C_{AS} + 1C_{CV} + 4C_{HM} + 2C_{SYM}$	–	$1C_{AS} + 1C_{HM} + 1C_{SYM} + 1C_{SV} + 1C_{DS}$	$1C_{AS} + 1C_{CV} + 1C_{SV} + 1C_{DS} + 8C_{HM} + 4C_{SYM}$
EAP-IKEv2	$3C_{SYM} + 1C_{DH} + 1C_{CV} + 1C_{DS} + 1C_{SV} + 1C_{HM}$	–	$3C_{SYM} + 1C_{DH} + 1C_{CV} + 1C_{DS} + 1C_{SV} + 1C_{HM}$	
HIP	$1C_{puzzle} + 1C_{SV} + 1C_{DH} + 1C_{DS} + 1C_{SYM} + 1C_{HM}$	$1C_{SV} + 1C_{DH} + 1C_{DS} + 1C_{SYM} + 1C_{HM}$	-	$1C_{puzzle} + 2C_{SV} + 2C_{DH} + 2C_{DS} + 2C_{SYM} + 2C_{HM}$
P2MP	$8C_{HM} + 1C_{DH}$	$6C_{HM} + 1C_{DH}$	$2C_{HM}$	$16C_{HM} + 2C_{DH}$

C_{SYM} cost for performing a symmetric encryption/decryption, C_{AS} cost for performing an asymmetric encryption/decryption, C_{DS} cost for performing a digital signature, C_{SV} cost for performing a signature validation, C_{DH} cost for performing a Diffie-Hellman operation, C_{HM} cost for performing a one-way HMAC function, C_{CV} cost for performing a certificate validation, C_{puzzle} cost for performing a puzzle–cryptographic challenge

Table 7 compares the communication overheads between the proposed and compared protocols in terms of roundtrip time. Although HIP has comparatively lower RTT. The proposed protocol is still recommended due to the fact that it sufficiently maintains all the security properties mentioned in Table 5.

6 Conclusions

Aiming to protect 5G wireless backhaul systems, it is highly needed to design a security protocol to realize security policy update, session key update, and balancing between efficiency and security in addition to meeting common security requirements. The security protocol is designed in such a way enables not only to deal with the diverse and personalized requirements for new 5G services and technologies, but also to defend against a number of attacks. Accordingly, with focus on such security requirements, the proposed protocol is devised to adjust its security policy and session key update based on the current network state and the network slice's security requirements. This assists the protocol to achieve adaptive security by dynamically changing cryptographic algorithms, authentication strength, key size, key lifetime, etc. Moreover, without further involvement of the authentication server, session key updates are carried out in periodical and efficient way, which plays a critical role in reducing information leakage and key compromise vulnerabilities in the backhaul network. The formal security analysis, using both BAN-logic and Scyther tool, proved that the protocol is secured against known attacks. Moreover, the security property comparison in contrast to the widely applied security standards (EAP-AKA, EAP-TLS, EAP-IKEv2, and HIP) indicates the proposed protocol utterly satisfies confidentiality, integrity, mutual authentication, key exchange, perfect forward secrecy, key update, policy update, and defense against resource exhaustion attack. Also, the computational overhead and communication overhead comparisons with the existing standards clearly show the proposed protocol performs better. Our proposed protocol takes into account current network conditions and reallocates security policies in the most effective way. The key advantage of the proposed protocol is its applicability in mobile backhaul links as an integrated part of backhaul networks. The proposed protocol can be utilized in real-time scenarios like 5G communication networks- which comprises of several hubs, multiple TMs as a ground terminal, and many end nodes in the forms of APs. For example, it can be applied to support security specific to network slices for special applications such as smart factories and smart healthcare. Our future works are three-folded. First, we will measure the actual performance overhead by implementing the proposed protocol. Second, we will study how to determine the security level in the backhaul system. Third, we will analyze how dynamic update of security policy affects the backhaul system in terms of performance and security.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1186/s13638-019-1592-0.

Additional file 1: Appendix. Source code for initial phase, key update phase, and policy update phase.

Abbreviations

AKA: Authentication and Key Agreement; ASF: Mobile backhaul authentication server function; BEET: Bounded End-to-End Tunnel; EAP-AKA: Extensible Authentication Protocol; ESP: Encapsulated Security Payload; FAP: Femto access points; HIP: Host Identity Protocol; HUB: Hub; IKEv1: Internet Key Exchange version 1; LTE: Long-Term Evolution; MOBIKE: IKEv2 Mobility and Multihoming; OpEx: Operating expenses; P2MP: Point-to-multipoint; PTP: Point-to-point; QoS: Quality of service; SSL: Secure socket layer; TLS: Transport layer security; TM: Terminal; V2X: Vehicle-to-Everything; VPN: Virtual private network

Authors' contributions

All authors have contributed to the work presented in this paper. Particularly, the contributions can be stated as follows: IY, JK, GC, JH, and DG contributed to the problem formulation, protocol designing, writing, and formal analysis simulation. JK,IY, and DG contributed to the protocol design. JK, GC, and DG contributed to the formal analysis. JK, JH, and DG participated in the design of the study and performed the performance analysis. IY, JK, GC, JH, and DG contributed to writing the manuscript. Lastly, all authors read and approved the final manuscript.

Funding

This work was supported by 'The Cross-Ministry Giga KOREA Project' grant funded by the Korea government (MSIT) (no. GK18N0600, Development of 20 Gbps P2MP wireless backhaul for 5G convergence service) and by the Soonchunhyang University Research Fund.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analysed during the current study

Competing interests

The authors declare that they have no competing interests.

Received: 5 September 2019 Accepted: 30 October 2019
Published online: 04 December 2019

References

1. M. Jaber, M. A. Imran, R. Tafazolli, A. Tukmanov, 5g backhaul challenges and emerging research directions: A survey. IEEE Access. **4**, 1743–1766 (2016)
2. I. Mesogiti, E. Theodoropoulou, K. Filis, G. Lyberopoulos, A. Ropodi, K. Tsagkaris, P. Demestichas, N. Pleros, G. Kalfas, C. Vagionas, in *Paper presented at the 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. Fiber-wireless fronthaul/backhaul network architectures for 5g (IEEE, Barcelona, 2018), pp. 17–19
3. P. Amini, J. A. A. Emmanuel, U.S. Patent 10278179B2, Dedicated backhaul link for a robust wireless mesh network. Google Patents (2019). Patent App. 10/278, US, 179
4. X. Ge, S. Tu, G. Mao, V. Lau, L. Pan, Cost efficiency optimization of 5g wireless backhaul networks. IEEE Trans. Mob. Comput., 1–1 (2018). https://doi.org/10.1109/TMC.2018.2886897
5. M. Alzenad, M. Z. Shakir, H. Yanikomeroglu, M.-S. Alouini, Fso-based vertical backhaul/fronthaul framework for 5g+ wireless networks. IEEE Commun. Mag. **56**(1), 218–224 (2018)
6. U. Siddique, H. Tabassum, E. Hossain, D. I. Kim, Wireless backhauling of 5g small cells: challenges and solution approaches. IEEE Wirel. Commun. **22**(5), 22–31 (2015)

7. E. Lagunas, L. Lei, S. Chatzinotas, B. Ottersten, in *IEEE Wireless Communications and Networking Conference, Marrakech, Morocco, April 2019*. Power and flow assignment for 5g integrated terrestrial-satellite backhaul networks (IEEE, 2019), pp. 1–6. https://doi.org/10.1109/wcnc.2019.8885662
8. G. Choudhary, J. Kim, V. Sharma, Security of 5g-mobile backhaul networks: A survey. J. Wirel. Mob. Netw. Ubiquit. Comput. Dependable Appl. (JoWUA). **9**(4), 41–70 (2018)
9. H. T. Nguyen, N.-P. Nguyen, T. Q. Duong, W.-J. Hwang, in *Ultra-dense Networks for 5G and Beyond: Modelling, Analysis, and Applications*. Physical layer security for ultra-dense networks under unreliable backhaul connection, (2019), pp. 231–246. https://doi.org/10.1002/9781119473756.ch10
10. I. V. Kotenko, M. Kolomeets, A. Chechulin, Y. Chevalier, A visual analytics approach for the cyber forensics based on different views of the network traffic. JoWUA. **9**(2), 57–73 (2018)
11. T. Gupta, G. Choudhary, V. Sharma, A survey on the security of pervasive online social networks (posns). J. Internet Serv. Inf. Secur. (JISIS). **8**(2), 48–86 (2018)
12. Y.-C. Kao, J.-C. Liu, Y.-H. Wang, Y.-H. Chu, S.-C. Tsai, Y.-B. Lin, Automatic blocking mechanism for information security with sdn. J. Internet Serv. Inf. Secur. (JISIS). **9**(1), 60–73 (2019)
13. N. Renugadevi, C. Mala, Improved group key agreement for emergency cognitive radio mobile ad hoc networks. JoWUA. **6**(3), 73–86 (2015)
14. F. Campioni, S. Choudhury, F. Al-Turjman, Scheduling rfid networks in the iot and smart health era. J. Ambient Intell. Human Comput. **10**(4043), 1–15 (2019). https://doi.org/10.1007/s12652-019-01221-5
15. M. Liyanage, M. Ylianttila, A. Gurtov, A case study on security issues in lte backhaul and core networks. Case Studies in Secure Computing: Achievements and Trends. **1**, 167 (2014)
16. P. Nikander, J. Melen, A Bound End-to-End Tunnel (BEET) mode for ESP: draft-nikander-esp-beet-mode-09. Work in Progress (2007)
17. C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, *Internet key exchange protocol version 2 (ikev2)*. (RFC Editor, 2010). http://www.rfc-editor.org/rfc/rfc7296. Accessed 15 Sept 2019
18. P. Eronen, *Ikev2mobility andmultihoming protocol (mobike)*. (RFC Editor, 2006). http://www.rfc-editor.org/rfc/rfc4555. Accessed 15 Sept 2019
19. R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, *Host identity protocol*. (RFC Editor, 2008). http://www.rfc-editor.org/rfc/rfc7401. Accessed 15 Sept 2019
20. F. Al-Turjman, H. Zahmatkesh, L. Mostarda, Quantifying uncertainty in internet of medical things and big-data services using intelligence and deep learning. IEEE Access. **7**, 115749–115759 (2019)
21. V. Sharma, I. You, F.-Y. Leu, M. Atiquzzaman, Secure and efficient protocol for fast handover in 5g mobile xhaul networks. J. Netw. Comput. Appl. **102**, 38–57 (2018)
22. V. Sharma, Y. Ko, J. Kim, I. You, Security management for backhaul-aware 5g-v2x. arXiv preprint arXiv:1811.08273 (2018)
23. J. Arkko, H. Haverinen, *Extensible authentication protocolmethod for 3rd generation authentication and key agreement (eap-aka)*. (RFC Editor, 2006). http://www.rfc-editor.org/rfc/rfc4187. Accessed 15 Sept 2019
24. S. Namal, A. Gurtov, M. Bennis, in *Paper presented at the 2011 Future Network & Mobile Summit*. Securing the backhaul for mobile and multi-homed femtocells (IEEE, Warsaw, 2011), pp. 15–17
25. M. Liyanage, A. Gurtov, in *2012 IEEE Vehicular Technology Conference (VTC Fall), Quebec City, QC, Canada*. Secured vpn models for lte backhaul networks (IEEE, 2012), pp. 1–5. https://doi.org/10.1109/vtcfall.2012.6399037
26. M. Liyanage, P. Kumar, M. Ylianttila, A. Gurtov, Novel secure vpn architectures for lte backhaul networks. Secur. Commun. Netw. **9**(10), 1198–1215 (2016)
27. C. Gritti, M. Önen, R. Molva, W. Susilo, T. Plantard, Device identification and personal data attestation in networks. J. Wirel. Mob. Netw. Ubiquit. Comput. Dependable Appl. (JoWUA). **9**(4), 1–25 (2018)
28. G. Choudhary, V. Sharma, in *5G Enabled Secure Wireless Networks*. A survey on the security and the evolution of osmotic and catalytic computing for 5g networks (Springer, 2019), pp. 69–102. https://doi.org/10.1007/978-3-030-03508-2_3
29. D. Danny, A. C. Yao, On the security of public key protocols. IEEE Trans. Inf. Theory. **29**(2), 198–208 (1983)
30. M. Burrows, M. Abadi, R. M. Needham, A logic of authentication. Proc. R. Soc. Lond. A Math. Phys. Sci. **426**(1871), 233–271 (1989)
31. P. Syverson, I. Cervesato, in *Foundations of Security Analysis and Design, Lecture Notes in Computer Science*. The Logic of Authentication Protocols, vol. 2171 (Springer, Berlin, FOSAD 2000), pp. 63–137
32. C. J. Cremers, in *Computer Aided Verification. CAV 2008. Lecture Notes in Computer Science*. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols, vol. 5123 (Springer, Berlin, 2008), pp. 414–418. https://doi.org/10.1007/978-3-540-70545-1_38
33. I. You, Y. Hori, K. Sakurai, Enhancing svo logic for mobile ipv6 security protocols. JoWUA. **2**(3), 26–52 (2011)
34. I. You, J.-H. Lee, Spfp: Ticket-based secure handover for fast proxy mobile ipv6 in 5g networks. Comput. Netw. **129**, 363–372 (2017)
35. D. Shin, V. Sharma, J. Kim, S. Kwon, I. You, Secure and efficient protocol for route optimization in pmipv6-based smart home iot networks. IEEE Access. **5**, 11100–11117 (2017)
36. C. A. Meadows, in *Advances in Cryptology - ASIACRYPT'94. ASIACRYPT, Lecture Notes in Computer Science*. Formal verification of cryptographic protocols: A survey, vol. 917 (Springer, Berlin, 1994), pp. 133–150
37. C. Boyd, W. Mao, in *Advances in Cryptology - EUROCRYPT Š93. EUROCRYPT, Lecture Notes in Computer Science*. On a Limitation of BAN Logic, vol. 765 (Springer, Berlin, 1993), pp. 240–247
38. C. Cremers, S. Mauw, in *Operational Semantics and Verification of Security Protocols, Information Security and Cryptography*. Chapter 4: Security Properties (Springer-Verlag, Berlin, 2012), pp. 37–65
39. D. Simon, B. Aboba, R. Hurst, *The eap-tls authentication protocol*. (RFC Editor, 2008). http://www.rfc-editor.org/rfc/rfc5216.txt. Accessed 15 Sept 2019
40. H. Tschofenig, D. Kroeselberg, A. Pashalidis, Y. Ohba, F. Bersani, *The extensible authentication protocol-internet key exchange protocol version 2 (eap-ikev2) method*. (RFC Editor, 2008). http://www.rfc-editor.org/rfc/rfc5106.txt. Accessed 15 Sept 2019
41. P. Nikander, A. Gurtov, T. R. Henderson, Host identity protocol (hip): connectivity, mobility, multi-homing, security, and privacy over ipv4 and ipv6 networks. IEEE Commun. Surv. Tutorials. **12**(2), 186–204 (2010)

Publisher's Note