This is the full version of the paper which appears in:
Information Security Applications: 10th International Workshop, WISA 2009, Busan, Korea, August 25-27, 2009.
H. Y. Youm, M. Yung (Eds.) Springer-Verlag, LNCS vol. ???? , pp. ??-??.

Securing Remote Access Inside Wireless Mesh Networks

(Full Version)

Mark Manulis

Cryptographic Protocols Group
Department of Computer Science
TU Darmstadt & CASED, Germany
mark@manulis.eu

Abstract. Wireless mesh networks (WMNs) that are being increasingly deployed in communities and public places provide a relatively stable routing infrastructure and can be used for diverse carrier-managed services. As a particular example we consider the scenario where a mobile device initially registered for the use with one wireless network (its home network) moves to the area covered by another network inside the same mesh. The goal is to establish a secure access to the home network using the infrastructure of the mesh.

Classical mechanisms such as VPNs can protect end-to-end communication between the mobile device and its home network while remaining transparent to the routing infrastructure. In WMNs this transparency can be misused for packet injection leading to the unnecessary consumption of the communication bandwidth. This may have negative impact on the cooperation of mesh routers which is essential for the connection establishment.

In this paper we describe how to establish remote connections inside WMNs while guaranteeing secure end-to-end communication between the mobile device and its home network *and* secure transmission of the corresponding packets along the underlying multi-hop path. Our solution is a provably secure, yet lightweight and round-optimal remote network access protocol in which intermediate mesh routers are considered to be part of the security architecture. We also sketch some ideas on the practical realization of the protocol using known standards and mention extensions with regard to forward secrecy, anonymity and accounting.

1 Introduction and Motivation

The increasing deployment of wireless networks in urban areas set up in private households and in public places offers high potential for ubiquitous communication to mobile clients. A promising approach for combining the power of different wireless networks, possibly under distinct administrative control, while expanding their overall coverage area is the composition into a *wireless mesh network (WMN)* through the deployment of dynamic routing protocols such as AODV [24], DSR [18], LQSR [13]. In this way a WMN consists of individual nodes (routers) which communicate with each other over wireless links in a multi-hop fashion. Many current WMNs are based on WLAN standards (IEEE 802.11a/b/g) so that mesh routing is performed at the network layer. A more efficient WMN infrastructure with mesh routing at the link layer is currently being specified within IEEE 802.11s. In contrast to a more general form of ad-hoc networks WMNs enjoy *static* topology with mostly stable routing infrastructure. This infrastructure allows for various carrier-managed services, which may also apply amongst wireless networks that compose the *same* mesh. One of them is the remote network access.

A typical urban WMN connects not only residential networks but also networks of institutions of public interest such as civil administration, police station, doctor's office/surgery, etc. Such networks (which we refer to as *home networks*) can be accessed by employees through the mobile devices initialized for this purpose while being in the proximity of the corresponding access points. On the other hand, it is desirable to allow remote access to these networks from locations covered by other wireless networks, which are still part of the same mesh. For example, doctors visiting patients at home or employees of a local health authority while inspecting restaurants in the neighborhood may wish to remotely access databases and application servers in their institutions in order to perform tasks that they would usually do within their native environment. Observe that these scenarios do not assume high mobility of the clients and the established WMN routing path will likely remain unchanged for the whole duration of the session.

An important security goal of a remote network access protocol is to protect the end-to-end communication between the device and its home network, in particular to protect the application content from being eavesdropped or modified during its transmission. Another significant problem in WMNs is that intermediate mesh routers are under different administrative control and that their cooperation is crucial for the establishment and maintenance of the remote session. A promising approach to increase this cooperation and so improve the robustness of the remote connection is to deploy additional incentive mechanisms [9], e.g. let the home network reimburse the cooperating mesh routers for resources that they allocate to establish and maintain the remote session. It is clear that traditional end-to-end security mechanisms (such as IPsec VPNs) that are transparent to the underlying routing infrastructure do not adequately reflect the additional security needs of cooperating mesh routers along the remote path.

1.1 Refining End-to-End Protection with the Concept of Path Security

With traditional authentication and key establishment protocols for end-to-end security the mobile device and its home network can compute the secret *end-to-end key* for the session after authenticating each other. This key is sufficient to protect the application data exchanged between the device and its remotely accessed home network.

Nevertheless, the communication nature of WMNs allows the adversary to inject packets into the wireless channel and although the deployed end-to-end protection will prevent the end points from accepting such rogue packets the cooperating mesh routers on the path will not be able to distinguish them from the "good" packets originated by the end points. As a consequence, mesh routers will forward rogue packets, thus wasting their own (costly) communication resources. This illustrates the need of stronger protection mechanisms that would ensure authentication at the packet level as well.

We observe that the above problem comes from the missing binding between the end-to-end protected (application) channel and the underlying multi-hop WMN channel. In this work we investigate how to efficiently bind these independent channels for the duration of the remote session.[1] Our solution is to refine the traditional concept of end-to-end security by achieving similar goals for the underlying path (*path security*). More precisely, we provide all entities involved into the remote session, i.e. the mobile device, its home network, and intermediate mesh routers, with the ability to compute an additional *path key* as a by-product of an authentication protocol for the establishment of the end-to-end protected remote connection. In order to ensure security of this path key we also require authentication of intermediate mesh routers and the end points during the execution of this protocol. In this way the application content exchanged between the mobile device and its home network can be still protected using the end-to-end key whereas the path key will protect the multi-hop transmission of corresponding packets along the established path, thus thwarting possible resource consumption attacks on the mesh routers. The existence of the path key shared amongst the end points and the mesh routers may also offer further application-specific benefits once the remote connection is established. For example, it can be used to protect control messages used in quality-of-service (QoS) mechanisms in some real-time applications such as VoIP.

[1] We remark that for networks with frequent route updates and for applications with highly mobile clients it would be rather undesirable to bind the application session to the underlying routing path. Yet, *inside* WMNs where the routes are rather stable and for remote sessions in which the clients remain static (as in our application examples) this binding is a promising practical method to provide stronger security guarantees for the remote connection.

1.2 Related Work

The security architecture for WLANs specified in IEEE 802.11i provides an authenticated network access of a mobile device to a wireless network with additional computation of a session key. The authentication can be based on pre-shared keys (WPA, WPA2) or use the IEEE 802.1X specification based on the EAP framework [2], the latter is often used in combination with a RADIUS authentication server. However, IEEE 802.11i has been designed for single-hop connections and is, therefore, not directly applicable for WMNs. A direct extension of IEEE 802.11i to a multi-hop scenario has been described in [22]. Further, with PANA [14] there exists an authentication carrying protocol that can encapsulate EAP messages and transmit them in a multi-hop fashion across WMNs as described in [10, 19]. In this way any EAP method, e.g. using username/passwords [11], shared keys [7], or public-key certificates [17, 25], can be applied. Alternatively, end-to-end security between the mobile client and its home network can be established using IPsec VPN tunnels.

The previously mentioned solutions, especially using current EAP methods, have been designed to achieve the classical end-to-end security goals. Therefore, they do not include any mechanisms that would be sufficient for the requirements of path security. In particular, they offer no protection against the impersonation of mesh routers and resource consumption attacks. On the other hand, with LHAP [29] there exists a light-weight authentication protocol (based on one-way hash functions) that can be used to obtain packet authentication in multi-hop communication scenarios. But LHAP was not designed to establish an end-to-end secure connection which is inherent to our setting. Moreover, LHAP does not offer mutual authentication between the mesh routers and the accessed home network.

Finally, we mention that the notion of path authentication is frequently used in the literature on routing protocols, especially in the context of secure route updates and announcements [26–28]. However, this process is oblivious to the actual application and is therefore not applicable in our case.

1.3 Organization

In Section 2 we describe our remote network access protocol, called SERENA, that simultaneously achieves the requirements of end-to-end and path security through the corresponding binding of the application channel to the underlying mesh path. In addition we provide some remarks on its efficiency and describe ideas on how to realize the protocol in practice using known authentication standards. In Section 3 we specify a single formal model for a secure remote network access protocol inside the wireless mesh capturing the mutual authentication and key establishment goals within the concepts of end-to-end and path security. The concept of path security is thereby modeled through the consideration of mesh routers as an inherent part of the security architecture. We utilize the classical modeling techniques from [5] which we extend to the multi-hop setting of WMNs. Following the model, in Section 4 we formally argue on the security of our protocol. Finally, in Section 5 we briefly discuss some further extensions regarding forward secrecy of the computed keys, anonymity of the mobile client with respect to the mesh network infrastructure, and accounting between the networks.

2 SERENA: A Single Protocol for End-to-End and Path Security

Here we introduce SERENA — our protocol for the secure remote access of the mobile device $\mathcal{M}$ to its wireless home network $\mathcal{H}$ using intermediate mesh routers $\{\mathcal{R}_i\}_i$, all belonging to the same mesh.

2.1 Notations and Building Blocks

SERENA uses several (well-known) cryptographic primitives:

- A *pseudo-random function* $\mathtt{PRF} : \{0,1\}^\kappa \times \{0,1\}^* \rightarrow \{0,1\}^*$ which is used for the purpose of key derivation and can be realized using block-ciphers or keyed one-way hash functions. By $\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa)$ we denote the maximum advantage over all PPT adversaries (running within time κ) in distinguishing the outputs of PRF from those of a random function better than by a random guess. We use PRF to derive various keys and sometimes we assume that the output is split in two parts, e.g. PRF with *expansion* such as the one defined within the TLS standard and analyzed in [15] can be used for this purpose.
- An *asymmetric encryption scheme* satisfying the property of indistinguishability under (adaptive) chosen-ciphertext attacks (IND-CCA2) whose encryption and decryption operations are denoted Enc and Dec, respectively. By $\mathsf{Adv}_{(\mathtt{Enc},\mathtt{Dec})}^{\mathtt{ind\text{-}cca2}}(\kappa)$ we denote the maximum advantage over all PPT adversaries (running within time κ) in breaking the IND-CCA2 property of $(\mathtt{Enc}, \mathtt{Dec})$ better than by a random guess; The property of IND-CCA2 security is for example preserved in several encryption schemes including RSA-OAEP [16] and DHIES [1].
- A *digital signature scheme* with existential unforgeability under chosen message attacks (EUF-CMA) whose signing and verification operations are denoted Sig and Ver, respectively. By $\mathsf{Succ}_{(\mathtt{Sig},\mathtt{Ver})}^{\mathtt{euf\text{-}cma}}(\kappa)$ we denote the maximum success probability over all PPT adversaries (running within time κ) given access to the signing oracle in finding a forgery; examples of such schemes include DSS [23] and PSS [6].
- A *sequential aggregate signature scheme* with existential unforgeability under chosen message attacks (EUF-CMA) whose aggregate signing and verification operations are denoted ASig and AVer, respectively. By $\mathsf{Succ}_{(\mathtt{ASig},\mathtt{AVer})}^{\mathtt{euf\text{-}cma}}(\kappa)$ we denote the maximum success probability over all PPT adversaries (running within time κ) given access to the aggregate signing oracle of an uncorrupted signer in finding an aggregate signature forgery; for formal security definitions and construction examples we refer to [20,21].
- A *message authentication code* function MAC with *weak unforgeability against chosen message attacks* (WUF-CMA) [4], e.g., the popular function HMAC [3] can be used for this purpose. By $\mathsf{Succ}_{\mathtt{MAC}}^{\mathtt{wuf\text{-}cma}}(\kappa)$ we denote the maximum success probability over all PPT adversaries (running within time κ) given access to the MAC oracle in finding a MAC forgery.

2.2 Initialization of SERENA

We assume that prior to the execution of Π the involved parties are in possession of the following long-lived keys: LL_i of each mesh router $\mathcal{R}_i$ consists of a private/public signature/verification key pair (sk_i, vk_i) which is suitable for the deployed aggregate signature scheme $(\mathtt{ASig}, \mathtt{AVer})$ and a decryption/encryption key pair (dk_i, ek_i); $LL_{\mathcal{H}}$ consists of a private/public signature/verification key pair $(sk_{\mathcal{H}}, vk_{\mathcal{H}})$ which is suitable for the digital signature scheme $(\mathtt{Sig}, \mathtt{Ver})$ and a pair $(\mathcal{M}, (k_{\mathcal{M}}, \alpha_{\mathcal{M}}))$ where $(k_{\mathcal{M}}, \alpha_{\mathcal{M}})$ is a high-entropy secret key consisting of a PRF key $k_{\mathcal{M}}$ and a MAC key $\alpha_{\mathcal{M}}$ shared with the hosted $\mathcal{M}$; $LL_{\mathcal{M}}$ consists of $(k_{\mathcal{M}}, \alpha_{\mathcal{M}})$. In practice it is sufficient for $\mathcal{H}$ and $\mathcal{M}$ to share $k_{\mathcal{M}}$ and derive the corresponding MAC key $\alpha_{\mathcal{M}}$ as $\mathtt{PRF}_{k_{\mathcal{M}}}(l)$ for some publicly fixed label l. Further, we assume that the public keys of all routers of the same mesh (including the home network $\mathcal{H}$) are known. Note that due to the static infrastructure of wireless mesh networks this assumption is feasible (this is clearly in contrast to (dynamic) ad-hoc networks where such assumptions are undesirable).

2.3 Lightweight SERENA — Specification

The detailed specification of lightweight SERENA is illustrated in Figure 1. For the clarity of presentation we assume that the path between $\mathcal{M}$ and $\mathcal{H}$ through the mesh network is given by a sequence of intermediate mesh routers $\mathcal{R}_1, \ldots, \mathcal{R}_n$ whereby each $\mathcal{R}_i$ knows the next hop router $\mathcal{R}_{i+1}$ as a result of underlying routing mechanism. In the following we highlight how SERENA merges end-to-end protection with path security.

SERENA uses distinct *labels* l_i, $i = 1,\ldots,4$, which are fixed in advance, as input to PRF for the derivation of different keys. Also, during the execution each party $P \in \{\mathcal{M}, \{\mathcal{R}_i\}_i, \mathcal{H}\}$ computes own session id sid_P as a concatenated bit string $\mathcal{M}|r_{\mathcal{M}}|\{\mathcal{R}_i|r_i\}_i|\mathcal{H}|r_{\mathcal{H}}$ where r_P denotes a *random nonce* chosen by P. For this parties use auxiliary variables $usid$ and $dsid$ which contain respective substrings of concatenated identities and nonces, depending on the position of P in the path.

Both $\mathcal{H}$ and $\mathcal{M}$ derive their end-to-end key K_e using the shared key $k_{\mathcal{M}}$ and their session ids. The equality of their session ids is ensured upon the verification of the corresponding MAC values $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$, whereby bits 0 and 1 are used to guarantee that $\mu_{\mathcal{H}} \neq \mu_{\mathcal{M}}$.

The difficult part of the protocol from the security point of view is the computation of the path key K_p and the mutual authentication between $\mathcal{H}$ and each mesh router $\mathcal{R}_i$. The main idea is to let the home network choose the secret key material k'_p which will be securely transported through the mesh network. Since the encryption of k'_p by $\mathcal{H}$ for each router $\mathcal{R}_i$ would lead to the linear increase of the ciphertext we apply a better approach — *hop-by-hop re-encryption*, i.e. k'_p originally encrypted by $\mathcal{H}$ for $\mathcal{R}_n$ is re-encrypted by $\mathcal{R}_n$ for $\mathcal{R}_{n-1}$ and so on. The challenge of such re-encryption is to allow each $\mathcal{R}_i$ to independently verify that decrypted k'_p is essentially the same as the one originally sent by $\mathcal{H}$, i.e. without trusting the next-hop router. The trick is to use the PRF output θ_p, which is never sent but is signed within $\sigma_{\mathcal{H}}$, which in turn is forwarded along the mesh path until it reaches $\mathcal{R}_1$. The verification of this $\sigma_{\mathcal{H}}$ implicitly requires each $\mathcal{R}_i$ to recompute θ_p and also to hold the same session id. Since θ_p and k_p (used to derive the path key K_p) are both derived from k'_p, the validity of the signature $\sigma_{\mathcal{H}}$ verified by $\mathcal{R}_i$ immediately implies the equality of k'_p received by $\mathcal{R}_i$ from $\mathcal{R}_{i+1}$ and the original k'_p sent by $\mathcal{H}$. Since the validity of $\sigma_{\mathcal{H}}$ also implies that $\mathcal{R}_i$ and $\mathcal{H}$ share the same session id, it is easy to see that at the end of the protocol each mesh router computes the same path key. At the same time the validity of $\sigma_{\mathcal{H}}$ is used for the authentication of $\mathcal{H}$ towards each $\mathcal{R}_i$. The mobile device $\mathcal{M}$ computes the same path key since it can derive k'_p directly using $k_{\mathcal{M}}$. The authentication of each $\mathcal{R}_i$ towards $\mathcal{H}$ is performed via the aggregate signature $\bar{\sigma}_{1..n}$. Each $\mathcal{R}_i$ contributes to the computation of this signature by executing the ASig algorithm using its own signing key sk_i, the aggregate signature $\bar{\sigma}_{1..i-1}$ previously received from $\mathcal{R}_{i-1}$, and the message $sid_i|\theta_p|\mu_{\mathcal{M}}$. The successful verification of $\bar{\sigma}_{1..n}$ by $\mathcal{H}$ authenticates the established path and ensures $\mathcal{H}$ that the mesh routers $\{\mathcal{R}_i\}_i$ hold the same session id and the path key as $\mathcal{H}$.

Note that during the protocol execution parties perform various checks. We assume that if at some point the corresponding check fails then the party immediately aborts. If parties do not abort then the execution of SERENA was successful, meaning that from now on $\mathcal{M}$ can be granted access to $\mathcal{H}$. In practice, $\mathcal{M}$ will be assigned an IP address from the domain of $\mathcal{H}$ and treated by $\mathcal{H}$ as one of its native devices.

2.4 Efficiency of SERENA

We stress that SERENA is lightweight in the sense that the mobile device $\mathcal{M}$ does *not* need to perform any costly public-key operations. Hence, SERENA can also be used with performance-constraint devices such as PDAs and smart phones provided they have a wireless IP interface. The public-key cryptography is used in SERENA for the hop-by-hop re-encryption of k'_p and the mutual authentication between $\{\mathcal{R}_i\}_i$ and $\mathcal{H}$. On the other hand, hop-by-hop re-encryption and the use of aggregate signatures allows to significantly reduce the (wireless) communication overhead which is the main bottleneck in WMNs (in contrast to the usually rich computation resources available to the mesh routers). Note also that SERENA requires three communication rounds — by one round we mean a complete message flow between $\mathcal{M}$ and $\mathcal{H}$ routed through $\{\mathcal{R}_i\}_i$, which is optimal to achieve mutual authentication (with key confirmation) based on the deployed challenge-response technique.

Remark 1. The modularity of SERENA allows to completely remove public-key operations (and the corresponding long-lived keys) resulting in a more efficient protocol that would nevertheless still ensure the

Mobile Device $\mathcal{M}$ $\{(k_\mathcal{M}, \alpha_\mathcal{M})\}$		Mesh Routers $\mathcal{R}_1, \dots, \mathcal{R}_n$ each $\mathcal{R}_i$ has $\{(sk_i, vk_i), (dk_i, ek_i)\}$		Home Network $\mathcal{H}$ $\{(sk_\mathcal{H}, vk_\mathcal{H}), (\mathcal{M}, (k_\mathcal{M}, \alpha_\mathcal{M}))\}$
$r_\mathcal{M} \in_r \{0,1\}^\kappa$; $usid_\mathcal{M} := \mathcal{M} \vert r_\mathcal{M}$				
	$\xrightarrow{usid_\mathcal{M} \vert \mathcal{H}}$			
		$r_i \in_r \{0,1\}^\kappa$ $usid_i := usid_{i-1} \vert \mathcal{R}_i \vert r_i$ SEND $usid_i$ to $\mathcal{R}_{i+1}$		
			$\xrightarrow{usid_n}$	
				$r_\mathcal{H} \in_r \{0,1\}^\kappa$ $sid_\mathcal{H} := usid_n \vert \mathcal{H} \vert r_\mathcal{H}$ $k'_p := \mathtt{PRF}_{k_\mathcal{M}}(l_1 \vert sid_\mathcal{H})$ $k_p \vert \theta_p := \mathtt{PRF}_{k'_p}(l_2 \vert sid_\mathcal{H})$ $\chi_n := \mathtt{Enc}_{ek_n}(k'_p)$ $\mu_\mathcal{H} := \mathtt{MAC}_{\alpha_\mathcal{M}}(0 \vert sid_\mathcal{H})$ $\sigma_\mathcal{H} := \mathtt{Sig}_{sk_\mathcal{H}}(sid_\mathcal{H} \vert \theta_p \vert \mu_\mathcal{H})$ $dsid_\mathcal{H} := H \vert r_\mathcal{H}$
			$\xleftarrow{dsid_\mathcal{H} \vert \chi_n \vert \mu_\mathcal{H} \vert \sigma_\mathcal{H}}$	
		$sid_i := usid_i \vert dsid_{i+1}$ $k'_p := \mathtt{Dec}_{dk_i}(\chi_{i+1})$ $k_p \vert \theta_p := \mathtt{PRF}_{k'_p}(l_2 \vert sid_i)$ CHECK $\sigma_\mathcal{H}$ $\chi_{i-1} := \mathtt{Enc}_{ek_{i-1}}(k'_p)$ $dsid_i := \mathcal{R}_i \vert r_i \vert dsid_{i+1}$ SEND $dsid_i \vert \chi_{i-1} \vert \mu_\mathcal{H} \vert \sigma_\mathcal{H}$ to $\mathcal{R}_{i-1}$		
	$\xleftarrow{dsid_1 \vert \mu_\mathcal{H}}$			
$sid_\mathcal{M} := usid_\mathcal{M} \vert dsid_1$ CHECK $\mu_\mathcal{H}$ $k'_p := \mathtt{PRF}_{k_\mathcal{M}}(l_1 \vert sid_\mathcal{M})$ $k_p \vert \alpha_p := \mathtt{PRF}_{k'_p}(l_2 \vert sid_\mathcal{M})$ $K_p := \mathtt{PRF}_{k_p}(l_3 \vert sid_\mathcal{M})$ $K_e := \mathtt{PRF}_{k_\mathcal{M}}(l_4 \vert sid_\mathcal{M})$ $\mu_\mathcal{M} := \mathtt{MAC}_{\alpha_\mathcal{M}}(1 \vert sid_\mathcal{M})$ ACCEPT				
	$\xrightarrow{\mu_\mathcal{M}}$			
		$\bar{\sigma}_{1..i} :=$ $\mathtt{ASig}_{sk_i}(\bar{\sigma}_{1..i-1}, sid_i \vert \theta_p \vert \mu_\mathcal{M})$ SEND $\mu_\mathcal{M} \vert \bar{\sigma}_{1..i}$ to $\mathcal{R}_{i+1}$ $K_p := \mathtt{PRF}_{k_p}(l_3 \vert sid_i)$ ACCEPT		
			$\xrightarrow{\mu_\mathcal{M} \vert \bar{\sigma}_{1..n}}$	
				CHECK $\mu_\mathcal{M}$ and $\bar{\sigma}_{1..n}$ $K_p := \mathtt{PRF}_{k_p}(l_3 \vert sid_\mathcal{H})$ $K_e := \mathtt{PRF}_{k_\mathcal{M}}(l_4 \vert sid_\mathcal{H})$ ACCEPT

Fig. 1. Execution of lightweight SERENA protocol between $\mathcal{M}$, $\mathcal{R}_1, \dots, \mathcal{R}_n$, and $\mathcal{H}$. Computations of intermediate mesh routers are described per each $\mathcal{R}_i$ and are triggered upon the delivery of the expected message from $\mathcal{R}_{i-1}$ or $\mathcal{R}_{i+1}$; indices $i = 0$ and $i = n + 1$ refer to $\mathcal{M}$ and $\mathcal{H}$, respectively.

traditional end-to-end security between $\mathcal{M}$ and $\mathcal{H}$. However, this modification will no longer allow for a secure computation of the path key and the mutual authentication between the home network and the mesh routers.

2.5 Implementation of SERENA using Authentication Standards

In order to realize SERENA with existing authentication standards we have to consider the following connection links: the wireless link between $\mathcal{M}$ and $\mathcal{R}_1$ and the wireless multi-hop path from $\mathcal{R}_1$ to $\mathcal{H}$ over the last-hop router $\mathcal{R}_n$.

We assume that $\mathcal{R}_1$ combines the routing functionality with that of an access point for $\mathcal{M}$. Prior to the execution of SERENA the mobile device $\mathcal{M}$ would be typically connected to $\mathcal{R}_1$ at the data link layer, which is preferable as no IP assignment for $\mathcal{M}$ is necessary in this case. Then, on the link between $\mathcal{M}$ and $\mathcal{R}_1$ our protocol can be implemented as a new EAP method within IEEE 802.1X framework. Along the multi-hop path from $\mathcal{R}_1$ to $\mathcal{H}$ (which combines the functionality of a mesh router and the authentication/application server) we can further implement SERENA using encapsulated EAP messages within an appropriate carrying protocol, for example with PANA [14].

Once SERENA is successfully executed $\mathcal{H}$ can allocate an IP address for $\mathcal{M}$ (either as a parameter within SERENA or via DHCP). This would complete the establishment of a secure remote network access between $\mathcal{M}$ and $\mathcal{H}$. In the course of subsequent communication between $\mathcal{M}$ and $\mathcal{H}$ intermediate mesh routers $\mathcal{R}_1, \ldots, \mathcal{R}_n$ can continue acting as layer 2 bridges until the session is finished.

The end-to-end protection of the session content between $\mathcal{M}$ and $\mathcal{H}$ can be achieved traditionally using the Authentication Header (AH) or Encapsulation Security Payload (ESP) mechanisms of IPsec in the *tunnel mode*, whereby the session key for this should be derived from K_e. The additional packet protection that should prevent resource consumption attacks via packet injection can be also performed using AH and ESP, however, this time in the *transport mode* and using the session key derived from K_p.

3 Security Model

Our model extends the two-party model from [5] towards the multi-hop communication nature of WMNs and the additional path security goals.

3.1 Participants and Communication

Mobile Device and Home Network The goal of the protocol, denoted abstractly by Π, is to establish a secure remote network access between the mobile device $\mathcal{M}$ and its wireless home network $\mathcal{H}$. The identity $\mathcal{M}$ is assumed to be unique within $\mathcal{H}$. Both $\mathcal{M}$ and $\mathcal{H}$ hold their corresponding long-lived keys $LL_{\mathcal{M}}$ and $LL_{\mathcal{H}}$. By $\kappa \in \mathbb{N}$ we denote the *security parameter* such that all secrets used in Π are polynomially bounded in κ.

Intermediate Mesh Routers We assume that $\mathcal{H}$ is part of the WMN and that the connection between $\mathcal{M}$ and $\mathcal{H}$ will be established along a multi-hop path consisting of some intermediate WMN routers $\mathcal{R}_1, \ldots, \mathcal{R}_n$. For the ease of presentation, we consider $\mathcal{R}_1$ as the first-hop router for $\mathcal{M}$ and $\mathcal{R}_n$ as the last-hop router before $\mathcal{H}$. Typically, such route will be defined in the connection establishment phase by the underlying routing protocol. Each intermediate mesh router $\mathcal{R}_i$ has a long-lived key LL_i.

Instances, Sessions, Partnering, Session Keys In order to model participation of $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ in distinct sessions of Π we consider an unlimited number of instances: By $[P, s]$ we denote the *s-th instance of* $P \in \{\mathcal{M}, \mathcal{H}, \{\mathcal{R}_i\}_i\}$ where $s \in \mathbb{N}$. Each instance $[P, s]$ is initialized with the party's long-lived key and invoked for one session which will be identified through a unique (public) *session id* sid_P^s. Instances of $\mathcal{M}$, $\mathcal{H}$ and $\{\mathcal{R}_i\}_i$ with identical session ids are *partnered*, i.e. participate in the same session.

Prior to the termination of Π an instance $[P, s]$ *accepts* if the protocol execution was successful (from the perspective of $[P, s]$) or *aborts*. A mesh router $\mathcal{R}_i$ accepts after it successfully authenticates $\mathcal{H}$ and computes the *session path key* $K_p \in \{0,1\}^\kappa$; the home network $\mathcal{H}$ accepts after it successfully authenticates the mobile device and the intermediate mesh routers on the path, and computes the *session end-to-end key* $K_e \in \{0,1\}^\kappa$ (in addition to K_p); and the mobile device $\mathcal{M}$ accepts after it successfully authenticates $\mathcal{H}$ and computes K_e and K_p.

3.2 Adversary and Security Definitions

Adversarial Model Adversary $\mathcal{A}$ is a PPT machine with complete control over the communication. It can mount attacks via the following set of queries:

- $\mathsf{Invoke}(P, m)$: This is the protocol invocation query for some entity $P \in \{\mathcal{M}, \mathcal{H}, \{\mathcal{R}_i\}_i\}$. In response, a new instance $[P, s]$ is created and its first protocol message is given to $\mathcal{A}$. The optional input m indicates the message expected by the instance to start the execution; for the initiator of the protocol m is supposed to be empty.
- $\mathsf{Send}(P, s, m)$: This query models communication control by $\mathcal{A}$ and contains a message m which should be delivered to the s-th instance of $P \in \{\mathcal{M}, \mathcal{H}, \{\mathcal{R}_i\}_i\}$. In response, $\mathcal{A}$ receives the outgoing message of $[P, s]$, or an empty string if $[P, s]$ terminates having processed m.
- $\mathsf{Corrupt}(P)$: This query models corruptions of $P \in \{\mathcal{M}, \mathcal{H}, \{\mathcal{R}_i\}_i\}$. In response, $\mathcal{A}$ receives LL_P.
- $\mathsf{Reveal\_Ke}(P, s)$: This query models independence of end-to-end keys computed by the instances of $P \in \{\mathcal{M}, \mathcal{H}\}$ in different sessions. In response, $\mathcal{A}$ is given K_e held by the instance; the query is answered only if $[P, s]$ has accepted.
- $\mathsf{Reveal\_Kp}(P, s)$: This query models independence of path keys computed by the instances of $P \in \{\mathcal{M}, \mathcal{H}, \{\mathcal{R}_i\}_i\}$ in different sessions. In response, $\mathcal{A}$ is given K_p held by the instance; the query is answered only if $[P, s]$ has accepted.
- $\mathsf{Test\_Ke}(P, s)$: This query will be used to define the AKE-security of the end-to-end key K_e and can be asked only for $P \in \{\mathcal{M}, \mathcal{H}\}$. It is answered only if $[P, s]$ has accepted and the answer is based on some secret bit $b \in \{0, 1\}$ chosen in advance: If $b = 1$ then $\mathcal{A}$ is given K_e, otherwise a randomly chosen value from $\{0,1\}^\kappa$.
- $\mathsf{Test\_Kp}(P, s)$: This query will be used to define the AKE-security of the path key K_p and can be asked for $P \in \{\mathcal{M}, \mathcal{H}, \{\mathcal{R}_i\}_i\}$. It is answered only if $[P, s]$ has accepted and the answer is based on some secret bit $b \in \{0, 1\}$ chosen in advance: If $b = 1$ then $\mathcal{A}$ is given K_p, otherwise a randomly chosen value from $\{0,1\}^\kappa$.

Definition 1 (Correctness). *A protocol Π is* correct *if the invoked instances of $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ terminate having accepted and* ***all*** *of the following holds: $\mathcal{M}$ and $\mathcal{H}$ hold the same end-to-end key K_e; $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ hold the same path key K_p.*

End-to-End Security Our first definition is about mutual authentication between $\mathcal{M}$ and $\mathcal{H}$. It also ensures that partnered instances of $\mathcal{M}$ and $\mathcal{H}$ accept holding the same end-to-end key K_e and the same path key K_p.

Definition 2 (MA between $\mathcal{M}$ and $\mathcal{H}$). *Given a correct protocol Π by $\mathsf{Game}_{\Pi}^{\mathsf{ma\text{-}m\text{-}h}}(\mathcal{A}, \kappa)$ we denote the interaction between the instances of $\mathcal{M}$, $\mathcal{H}$ and $\{\mathcal{R}_i\}_i$ with a PPT adversary $\mathcal{A}$ that is allowed to query* Invoke, Send, Corrupt, Reveal_Ke, *and* Reveal_Kp. $\mathcal{A}$ wins *if at some point during the interaction:*

(1) an uncorrupted instance of $\mathcal{M}$ accepts but there is ***no*** *uncorrupted partnered instance of $\mathcal{H}$,* ***or***

(2) an uncorrupted instance of $\mathcal{H}$ accepts but there is ***no*** *uncorrupted partnered instance of $\mathcal{M}$,* ***or***

(3) uncorrupted partnered instances of $\mathcal{M}$ and $\mathcal{H}$ accept without holding the same session end-to-end key K_e, ***or***

(4) $\{\mathcal{R}_i\}_i$ are uncorrupted and uncorrupted partnered instances of $\mathcal{M}$ and $\mathcal{H}$ accept without holding the same session path key K_p.

The maximum probability of this event over all adversaries (running in time κ) is denoted

$$\mathsf{Succ}_{\Pi}^{\text{ma-m-h}}(\mathcal{A},\kappa) = \max_{\mathcal{A}} |\Pr[\mathcal{A} \text{ wins in } \mathsf{Game}_{\Pi}^{\text{ma-m-h}}(\mathcal{A},\kappa)]|.$$

Π provides mutual authentication between $\mathcal{M}$ and $\mathcal{H}$ *if this probability is negligible in κ.*

The secrecy of the established end-to-end key K_e is modeled through the classical notion of authenticated key exchange (AKE) security adopted to our setting. Recall that the basic idea behind AKE-security is to require the indistinguishability of the key computed in some *test* session from some randomly chosen value by any *outsider* adversary. Observe that in case of K_e we must allow the adversary to corrupt the intermediate mesh routers $\{\mathcal{R}_i\}_i$ also during the test session.

In order to define the AKE-security of K_e (denoted as *e-AKE-security*) we first specify the auxiliary notion of *e-freshness* for the instances of $\mathcal{M}$ and $\mathcal{H}$. It provides conditions under which $\mathcal{A}$ can be treated as an outsider with respect to the test session for which it has to distinguish K_e.

Definition 3 (e-Freshness). *In the execution of Π an instance $[P,s]$ with $P \in \{\mathcal{M},\mathcal{H}\}$ is* e-fresh *if* ***none*** *of the following holds:*

- *$\mathcal{A}$ asks $\mathsf{Corrupt}(P)$;*
- *IF $P=\mathcal{M}$: $\mathcal{A}$ asks $\mathsf{Reveal\_Ke}(\mathcal{M},s)$ after $[\mathcal{M},s]$ has accepted or $\mathsf{Reveal\_Ke}(\mathcal{H},t)$ after $[\mathcal{H},t]$ has accepted and $[\mathcal{M},s]$ and $[\mathcal{H},t]$ are partnered;*
- *IF $P=\mathcal{H}$: $\mathcal{A}$ asks $\mathsf{Reveal\_Ke}(\mathcal{H},s)$ after $[\mathcal{H},s]$ has accepted or $\mathsf{Reveal\_Ke}(\mathcal{M},t)$ after $[\mathcal{M},t]$ has accepted and $[\mathcal{H},s]$ and $[\mathcal{M},t]$ are partnered.*

Definition 4 (e-AKE-Security). *Given a correct protocol Π, a uniformly chosen bit b, and a PPT adversary $\mathcal{A}$ with access to the queries* Invoke, Send, Corrupt, Reveal_Ke, Reveal_Kp, *and* Test_Ke, *by $\mathsf{Game}_{\Pi}^{\text{e-ake},b}(\mathcal{A},\kappa)$ we denote the following interaction between the instances of $\mathcal{M}$, $\mathcal{H}$ and $\{\mathcal{R}_i\}_i$ with $\mathcal{A}$:*

- *$\mathcal{A}$ interacts with instances via queries;*
- *at some point $\mathcal{A}$ asks a* Test_Ke *query to an instance $[P,s]$ which has accepted and is* ***e-fresh*** *(and remains such by the end of the interaction);*
- *$\mathcal{A}$ continues interacting with instances and when $\mathcal{A}$ terminates, it outputs a bit, which is then set as the output of the interaction.*

$\mathcal{A}$ wins *if the output of $\mathsf{Game}_{\Pi}^{\text{e-ake},b}(\mathcal{A},\kappa)$ is identical to b. The maximum probability of the adversarial advantage over the random guess of b, over all adversaries (running in time κ) is denoted*

$$\mathsf{Adv}_{\Pi}^{\text{e-ake}}(\mathcal{A},\kappa) = \max_{\mathcal{A}} |2\Pr[\mathsf{Game}_{\Pi}^{\text{e-ake},b}(\mathcal{A},\kappa) = b] - 1|.$$

If this advantage is negligible in κ then Π provides e-AKE-security.

Path Security Here we formalize the additional path security requirements from Section 1.1. First, we define mutual authentication between the home network $\mathcal{H}$ and each mesh router $\mathcal{R}_i$ on the path, which also captures the equality of session path keys computed by $\mathcal{H}$ and each $\mathcal{R}_i$. Note also that winning conditions in this case require the instance of $\mathcal{R}_i$ to be uncorrupted, however, the adversary can still corrupt any other $\mathcal{R}_j$, $j \neq i$. This models attacks of malicious routers aiming to disrupt the mutual authentication process and the computation of identical path keys between the (honest) intermediate mesh routers and the home network.

Definition 5 (MA between $\mathcal{H}$ and $\mathcal{R}_i$). *Given a correct protocol Π by* $\mathsf{Game}_{\Pi}^{\text{ma-h-r}}(\mathcal{A}, \kappa)$ *we denote the interaction between the instances of $\mathcal{M}$, $\mathcal{H}$ and $\{\mathcal{R}_i\}_i$ with a PPT adversary $\mathcal{A}$ that is allowed to query* Invoke, Send, Corrupt, Reveal_Ke, *and* Reveal_Kp. $\mathcal{A}$ wins *if at some point during the interaction:*

*(1) an uncorrupted instance of $\mathcal{R}_i$ accepts but there is **no** uncorrupted partnered instance of $\mathcal{H}$, **or***
*(2) an uncorrupted instance of $\mathcal{H}$ accepts but there is **no** uncorrupted partnered instance of $\mathcal{R}_i$, **or***
(3) $\mathcal{M}$ is uncorrupted and uncorrupted partnered instances of $\mathcal{H}$ and $\mathcal{R}_i$ accept without holding the same session path key K_p.

The maximum probability of this event over all adversaries (running in time κ) is denoted

$$\mathsf{Succ}_{\Pi}^{\text{ma-h-r}}(\mathcal{A}, \kappa) = \max_{\mathcal{A}} |\Pr[\mathcal{A} \text{ wins in } \mathsf{Game}_{\Pi}^{\text{ma-h-r}}(\mathcal{A}, \kappa)]|.$$

Π provides mutual authentication between $\mathcal{H}$ and $\mathcal{R}_i$ *if this probability is negligible in κ.*

Recall that the instances of participants are seen as partnered if they hold the same session ids. Hence, if Π provides both the mutual authentication between $\mathcal{M}$ and $\mathcal{H}$ *and* the mutual authentication between $\mathcal{H}$ and each $\mathcal{R}_i$ then the execution of Π in which $\mathcal{H}$ accepts implies the partnering for the corresponding instances of $\mathcal{M}$ and each $\mathcal{R}_i$, which in turn implies that the instances of $\mathcal{M}$, $\{\mathcal{R}_i\}_i$, and $\mathcal{H}$ computed the same K_p.

Further, we model the AKE-security of the path key K_p (denoted *p-AKE-security*) modeled using the auxiliary notion of *p-freshness* that specifies the conditions under which $\mathcal{A}$ can be treated as an outsider with respect to the test session for which it has to distinguish K_p.

Definition 6 (p-Freshness). *In the execution of Π an instance $[P, s]$ with $P \in \{\mathcal{M}, \mathcal{H}, \{\mathcal{R}_i\}_i\}$ is* p-fresh *if **none** of the following holds:*

- *$\mathcal{A}$ asks* Corrupt(P)*;*
- *If $P = \mathcal{M}$: $\mathcal{A}$ asks* Reveal_Kp$(\mathcal{M}, s)$ *after $[\mathcal{M}, s]$ has accepted or* Reveal_Kp(P', t) *for $P' \in \{\mathcal{H}, \{\mathcal{R}_i\}_i\}$ after $[P', t]$ has accepted and $[\mathcal{M}, s]$ and $[P', t]$ are partnered;*
- *If $P = \mathcal{H}$: $\mathcal{A}$ asks* Reveal_Kp$(\mathcal{H}, s)$ *after $[\mathcal{H}, s]$ has accepted or* Reveal_Kp(P', t) *for $P' \in \{\mathcal{M}, \{\mathcal{R}_i\}_i\}$ after $[P', t]$ has accepted and $[\mathcal{H}, s]$ and $[P', t]$ are partnered;*
- *If $P = \mathcal{R}_i$: $\mathcal{A}$ asks* Reveal_Kp$(\mathcal{R}_i, s)$ *after $[\mathcal{R}_i, s]$ has accepted or* Reveal_Kp(P', t) *for $P' \in \{\mathcal{M}, \mathcal{H}, \{\mathcal{R}_j\}_{j \neq i}\}$ after $[P', t]$ has accepted and $[\mathcal{R}_i, s]$ and $[P', t]$ are partnered.*

Definition 7 (p-AKE-Security). *Given a correct protocol Π, a uniformly chosen bit b, and a PPT adversary $\mathcal{A}$ with access to the queries* Invoke, Send, Corrupt, Reveal_Ke, Reveal_Kp, *and* Test_Kp, *by* $\mathsf{Game}_{\Pi}^{\text{p-ake},b}(\mathcal{A}, \kappa)$ *we denote the following interaction between the instances of $\mathcal{M}$, $\mathcal{H}$ and $\{\mathcal{R}_i\}_i$ with $\mathcal{A}$:*

- *$\mathcal{A}$ interacts with instances via queries;*
- *at some point $\mathcal{A}$ asks a* Test_Kp *query to an instance $[P, s]$ which has accepted and is **p-fresh** (and remains such by the end of the interaction);*
- *$\mathcal{A}$ continues interacting with instances and when $\mathcal{A}$ terminates, it outputs a bit, which is then set as the output of the interaction.*

$\mathcal{A}$ wins *if the output of* $\mathsf{Game}_{\Pi}^{\text{p-ake},b}(\mathcal{A},\kappa)$ *is identical to* b*. The maximum probability of the adversarial advantage over the random guess of* b*, over all adversaries (running in time* κ*) is denoted*

$$\mathsf{Adv}_{\Pi}^{\text{p-ake}}(\mathcal{A},\kappa) = \max_{\mathcal{A}} |2\Pr[\mathsf{Game}_{\Pi}^{\text{p-ake},b}(\mathcal{A},\kappa) = b] - 1|.$$

If this advantage is negligible in κ *then* Π *provides* p-AKE-security.

4 Security Analysis of SERENA

The following theorems show that SERENA provides both end-to-end and path security. By q we denote the total number of the invoked sessions.

Theorem 1 (MA between $\mathcal{M}$ and $\mathcal{H}$). *Given a WUF-CMA secure* MAC *the protocol* SERENA *specified in Figure 1 provides mutual authentication between the participating mobile device and its home network in the sense of Definition 2, and*

$$\mathsf{Succ}_{\mathtt{SERENA}}^{\text{ma-m-h}}(\kappa) \leq \frac{2q^2}{2^\kappa} + 2\mathsf{Succ}_{\mathtt{MAC}}^{\mathtt{wuf\text{-}cma}}(\kappa).$$

Proof. In our proofs we apply the meanwhile classical proving technique from [**?**]. Here we construct a *sequence of games* $\mathbf{G}_i$, $i = 0,\ldots,2$ and denote by $\mathsf{Win}_i^{\text{ma-m-h}}$ the event that an adversary $\mathcal{A}$ breaks the mutual authentication between $\mathcal{M}$ and $\mathcal{H}$ in game $\mathbf{G}_i$, i.e., wins in the corresponding interaction as described in Definition 2. Note that $\mathcal{A}$ is allowed to corrupt any mesh router $\mathcal{R}_i$ for the winning conditions (1) – (3), but not for (4).

Game $\mathbf{G}_0$. [*Real protocol*] This is the real $\mathsf{Game}_{\mathtt{SERENA}}^{\text{ma-m-h}}(\kappa)$ played between a simulator Δ and a PPT adversary $\mathcal{A}$. Δ simulates the actions of the participating $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ according to the protocol specification and answers all queries of $\mathcal{A}$.

Game $\mathbf{G}_1$. [*Collisions for chosen nonces* $r_{\mathcal{M}}$ *and* $r_{\mathcal{H}}$] In this game the simulation aborts if during the interaction the simulator chooses the same random nonce $r_{\mathcal{M}}$ resp. $r_{\mathcal{H}}$ on behalf of $\mathcal{M}$ resp. $\mathcal{H}$ in two different protocol sessions. Considering the collision probability for the same nonce to be chosen twice we obtain $|\Pr[\mathsf{Win}_1^{\text{ma-m-h}}] - \Pr[\mathsf{Win}_0^{\text{ma-m-h}}]| \leq \frac{2q^2}{2^\kappa}$.

Note that since in SERENA each participant computes own session id as a concatenated string $\mathcal{M}|r_{\mathcal{M}}|\{\mathcal{R}_i|r_i\}_i|\mathcal{H}|r_{\mathcal{H}}$ this game rules out the occurrence of the same session ids computed by the instances of $\mathcal{M}$ and $\mathcal{H}$ in two different sessions, regardless of the chosen $\{r_i\}_i$. Thus, this game implies that $sid_{\mathcal{M}}$ and $sid_{\mathcal{H}}$ remain unique for each invoked session.

Game $\mathbf{G}_2$. [*MAC forgeries for* $\mu_{\mathcal{H}}$ *and* $\mu_{\mathcal{M}}$] This game is identical to Game $\mathbf{G}_1$ with the only exception that Δ fails if $\mathcal{A}$ asks a Send query to an instance of $\mathcal{M}$ containing a valid MAC value $\mu_{\mathcal{H}}$ not previously output by an instance of $\mathcal{H}$ or a Send query to an instance of $\mathcal{H}$ containing a valid $\mu_{\mathcal{M}}$ not previously output by an instance of $\mathcal{M}$.

The probability that the simulation aborts can be upper-bounded through the probability of forging any of the both MAC values. To see this, consider Δ given access to the MAC oracle. Δ simulates the execution of SERENA according to the specification except that it computes $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$ through the corresponding oracle calls. In case that the simulation aborts Δ is in possession of a valid MAC value (representing either $\mu_{\mathcal{H}}$ or $\mu_{\mathcal{M}}$) which was not obtained through any previous oracle call. Hence, Δ can easily output it as a forgery. This implies $|\Pr[\mathsf{Win}_2^{\text{ma-m-h}}] - \Pr[\mathsf{Win}_1^{\text{ma-m-h}}]| \leq 2\mathsf{Succ}_{\mathtt{MAC}}^{\mathtt{wuf\text{-}cma}}(\kappa)$.

Having eliminated possible forgeries for $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$ we note that since these MAC values are computed over the session ids $sid_{\mathcal{H}}$ and $sid_{\mathcal{M}}$, respectively (that according to the previous game are unique for each new session) this game rules out any successful replay attacks using $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$. Therefore, any successful verification of $\mu_{\mathcal{H}}$ by an instance of $\mathcal{M}$ and of $\mu_{\mathcal{M}}$ by an instance of $\mathcal{H}$ implies that there are two instances of $\mathcal{M}$ and $\mathcal{H}$ that hold the same session ids, and are, therefore partnered. Since verification of $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$ is the necessary requirement for the acceptance of the instances of $\mathcal{M}$ and $\mathcal{H}$ in SERENA we follow that this game ensures mutual authentication between $\mathcal{M}$ and $\mathcal{H}$ and excludes attacks by which $\mathcal{A}$ can win based on conditions (1) and (2).

Further, we focus on the attacks based on conditions (3) and (4). Since the session end-to-end key K_e is derived by the instances of $\mathcal{M}$ and $\mathcal{H}$ in a deterministic way as $\mathtt{PRF}_{k_{\mathcal{M}}}(l_4, sid_{\mathcal{M}})$ and $\mathtt{PRF}_{k_{\mathcal{M}}}(l_4, sid_{\mathcal{H}})$, respectively, it follows that if any two partnered instances of $\mathcal{M}$ and $\mathcal{H}$ accept then they hold identical values for K_e, i.e. the probability of $\mathcal{A}$ to win in this game through condition 3 is 0. Finally, we observe that if any two partnered instances of $\mathcal{M}$ and $\mathcal{H}$ accept then they hold the path key K_p, which is computed in a deterministic way as $\mathtt{PRF}_{k_p}(l_3, sid_{\mathcal{M}})$ and $\mathtt{PRF}_{k_p}(l_3, sid_{\mathcal{H}})$, respectively. Hence, the probability that $\mathcal{A}$ wins in this game through condition 4, is upper-bounded by the probability that these partnered instances have computed different values for the key material k_p. However, k_p is derived in two deterministic steps using the shared secret $k_{\mathcal{M}}$, i.e. $\mathcal{M}$ computes $k'_p := \mathtt{PRF}_{k_{\mathcal{M}}}(l_1, sid_{\mathcal{M}})$ followed by $k_p|\alpha_p := \mathtt{PRF}_{k'_p}(l_1, sid_{\mathcal{M}})$ whereas $\mathcal{H}$ computes $k'_p := \mathtt{PRF}_{k_{\mathcal{M}}}(l_1, sid_{\mathcal{H}})$ followed by $k_p|\alpha_p := \mathtt{PRF}_{k'_p}(l_1, sid_{\mathcal{H}})$. This implies that the probability of $\mathcal{A}$ winning in this game through condition 4 is also 0. Summing up the discussed probabilities of $\mathcal{A}$ to win in this game based on conditions 3 and 4 we obtain $\Pr[\mathsf{Win}_2^{\mathsf{ma\text{-}m\text{-}h}}] = 0$. Combining with the previous equations, we conclude the proof.

Theorem 2 (e-AKE). *Given a WUF-CMA secure* MAC *and a pseudo-random* PRF *the protocol* SERENA *specified in Figure 1 provides e-AKE-security in the sense of Definition 4, and*

$$\mathsf{Adv}_{\mathtt{SERENA}}^{\mathsf{e\text{-}ake}}(\kappa) \leq \frac{4q^2}{2^\kappa} + 4\mathsf{Succ}_{\mathtt{MAC}}^{\mathtt{wuf\text{-}cma}}(\kappa) + 4q\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa).$$

Proof. As in the previous proof we construct a sequence of games $\mathbf{G}_i$, $i = 0, \ldots, 4$ and denote by $\mathsf{Win}_i^{\mathsf{e\text{-}ake}}$ the event that $\mathcal{A}$ breaks the e-AKE-security of SERENA in game $\mathbf{G}_i$, i.e., wins in the corresponding interaction as described in Definition 4.

Game $\mathbf{G}_0$. [*Real protocol*] This is the real $\mathsf{Game}_{\mathtt{SERENA}}^{\mathsf{e\text{-}ake}}(\kappa)$ played between a simulator Δ and a PPT adversary $\mathcal{A}$. Δ simulates the actions of the participating $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ according to the protocol specification and answers all queries of $\mathcal{A}$. Recall, that the Test_Ke query is asked by $\mathcal{A}$ to an e-fresh instance of either $\mathcal{M}$ or $\mathcal{H}$ which has previously accepted. In order to prevent $\mathcal{A}$ from active participation on behalf of either $\mathcal{M}$ or $\mathcal{H}$ we first exclude possible impersonation attacks against any of these parties. For this we can re-use games $\mathbf{G}_1$ and $\mathbf{G}_2$ from the proof of Theorem 1.

Game $\mathbf{G}_1$. [*Collisions for nonces $r_{\mathcal{M}}$ and $r_{\mathcal{H}}$*] The simulation in this game aborts (and the output bit of the interaction is set at random) if the same random nonce $r_{\mathcal{M}}$ (or $r_{\mathcal{H}}$) is chosen by Δ on behalf of $\mathcal{M}$ (or $\mathcal{H}$) in two different protocol sessions, implying

$$|\Pr[\mathsf{Win}_1^{\mathsf{e\text{-}ake}}] - \Pr[\mathsf{Win}_0^{\mathsf{e\text{-}ake}}]| \leq \frac{2q^2}{2^\kappa}. \tag{1}$$

Game $\mathbf{G}_2$. [*MAC forgeries for $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$*] The simulation in this game aborts (and the output bit of the interaction is set at random) if $\mathcal{A}$ asks as part of its Send query to $\mathcal{M}$ (or to $\mathcal{H}$) a valid MAC value $\mu_{\mathcal{H}}$ (or $\mu_{\mathcal{M}}$) which was not previously output by an instance of $\mathcal{H}$ (or $\mathcal{M}$), so that

$$|\Pr[\mathsf{Win}_2^{\mathsf{e\text{-}ake}}] - \Pr[\mathsf{Win}_1^{\mathsf{e\text{-}ake}}]| \leq 2\mathsf{Succ}_{\mathtt{MAC}}^{\mathtt{wuf\text{-}cma}}(\kappa). \tag{2}$$

The elimination of possible forgeries and replay attacks for $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$ implies that any accepting e-fresh instance of $\mathcal{M}$ (or $\mathcal{H}$) has a partnered instance of $\mathcal{H}$ (or $\mathcal{M}$) which is also e-fresh.

Game $\mathbf{G}_3$. [*Pseudo-randomness of k'_p*] This game is identical to Game $\mathbf{G}_2$ except that Δ in each session chooses the value for k'_p at random on behalf of an instance of $\mathcal{H}$ instead of deriving it via PRF and uses the same value in the simulation of the corresponding partnered instance of $\mathcal{M}$ (to preserve consistency). Following the classical reductionist argument we obtain

$$|\Pr[\mathsf{Win}_3^{\mathsf{e\text{-}ake}}] - \Pr[\mathsf{Win}_2^{\mathsf{e\text{-}ake}}]| \leq q\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa). \tag{3}$$

Note that this game ensures independence between k'_p (and consequently between all secrets that are derived using k'_p such as k_p, α_p, and the path key K_p) and the shared key $k_{\mathcal{M}}$ (used to derive the end-to-end key K_e). We remark that the randomly chosen value for k'_p in this game will also be propagated via hop-by-hop encryption across the mesh network in order to ensure consistency between the path key computed by the possibly corrupted instances of $\{\mathcal{R}_i\}_i$ and uncorrupted instances of $\mathcal{H}$ and $\mathcal{M}$.

Game $\mathbf{G}_4$. [*Pseudo-randomness of K_e*] This final game is identical to Game $\mathbf{G}_3$ except that Δ in each session chooses K_e at random on behalf of an instance of $\mathcal{M}$ instead of deriving it via $\mathtt{PRF}$ and uses the same value in the simulation of the corresponding partnered instance of $\mathcal{H}$ (to preserve consistency), so that

$$|\Pr[\mathsf{Win}_4^{\text{e-ake}}] - \Pr[\mathsf{Win}_3^{\text{e-ake}}]| \le q\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa). \tag{4}$$

This game implies that the answer given to $\mathcal{A}$ in response to its Test_Ke query to some e-fresh instance of $\mathcal{M}$ or $\mathcal{H}$ is a completely random value, regardless of the chosen bit b. Obviously, the probability of $\mathcal{A}$ to win in this game is then given by the probability of a random guess, i.e. $\Pr[\mathsf{Win}_4^{\text{e-ake}}] = \frac{1}{2}$. The combination of the above equations concludes the proof.

Theorem 3 (MA between $\mathcal{H}$ and $\mathcal{R}_i$). *Given EUF-CMA secure $(\mathtt{Sig}, \mathtt{Ver})$ and $(\mathtt{ASig}, \mathtt{AVer})$ the protocol $\mathtt{SERENA}$ specified in Figure 1 provides mutual authentication between each participating mesh router $\mathcal{R}_i$ and the home network $\mathcal{H}$ in the sense of Definition 5, and*

$$\mathsf{Succ}_{\mathtt{SERENA}}^{\text{ma-h-r}}(\kappa) \le \frac{(n+1)q^2}{2^\kappa} + \mathsf{Succ}_{(\mathtt{Sig},\mathtt{Ver})}^{\mathtt{euf\text{-}cma}}(\kappa) + n\mathsf{Succ}_{(\mathtt{ASig},\mathtt{AVer})}^{\mathtt{euf\text{-}cma}}(\kappa).$$

Proof. Similar to the previous proofs we construct a sequence of games $\mathbf{G}_i$, $i = 0, \ldots, 2$ and denote by $\mathsf{Win}_i^{\text{ma-h-r}}$ the event that $\mathcal{A}$ breaks the mutual authentication between any $\mathcal{R}_i$ and $\mathcal{H}$ in game $\mathbf{G}_i$, i.e., wins in the corresponding interaction as described in Definition 5. Note that $\mathcal{A}$ is allowed to corrupt $\mathcal{M}$ for the winning conditions (1) and (2), but not for (3).

Game $\mathbf{G}_0$. [*Real protocol*] This is the real $\mathsf{Game}_{\mathtt{SERENA}}^{\text{ma-h-r}}(\kappa)$ played between a simulator Δ and a PPT adversary $\mathcal{A}$. Δ simulates the actions of the participating $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ according to the protocol specification and answers all queries of $\mathcal{A}$.

Game $\mathbf{G}_1$. [*Collisions for nonces r_i and $r_{\mathcal{H}}$*] Similar to the proof of Theorem 1 we abort the simulation in this game if during the interaction Δ chooses the same random nonce r_i resp. $r_{\mathcal{H}}$ on behalf of any $\mathcal{R}_i$ resp. $\mathcal{H}$ in two different protocol sessions. Considering at most n mesh routers we obtain

$$|\Pr[\mathsf{Win}_1^{\text{ma-h-r}}] - \Pr[\mathsf{Win}_0^{\text{ma-h-r}}]| \le \frac{(n+1)q^2}{2^\kappa}. \tag{5}$$

Obviously, this game implies that each sid_i and $sid_{\mathcal{H}}$ remains unique for each invoked session, regardless of the chosen $r_{\mathcal{M}}$.

Game $\mathbf{G}_2$. [*Signature forgery for $\sigma_{\mathcal{H}}$*] This game is identical to Game $\mathbf{G}_1$ with the only exception that Δ fails if $\mathcal{A}$ asks a Send query to an instance of any $\mathcal{R}_i$ containing a valid signature $\sigma_{\mathcal{H}}$ not previously output by an instance of $\mathcal{H}$.

Assume that Δ simulates the protocol execution according to the specification except that it is given access to the signing oracle (representing the algorithm $\mathtt{Sig}$) which it queries in order to obtain the corresponding signatures $\sigma_{\mathcal{H}}$ on behalf of $\mathcal{H}$. In case that the simulation aborts Δ is in possession of a valid signature $\sigma_{\mathcal{H}}$ which was not obtained through any previous oracle call, and can, therefore, be returned by Δ as a corresponding forgery. Hence, $|\Pr[\mathsf{Win}_2^{\text{ma-h-r}}] - \Pr[\mathsf{Win}_1^{\text{ma-h-r}}]| \le \mathsf{Succ}_{(\mathtt{Sig},\mathtt{Ver})}^{\mathtt{euf\text{-}cma}}(\kappa)$.

Since each $\sigma_{\mathcal{H}}$ is computed over the corresponding session id $sid_{\mathcal{H}}$ and verified by $\mathcal{R}_i$ using own sid_i (amongst other inputs) and the session ids are unique according to the previous game, this game rules out any successful replay attacks for $\sigma_{\mathcal{H}}$. Since verification of $\sigma_{\mathcal{H}}$ by each $\mathcal{R}_i$ is the necessary requirement for the acceptance of $\mathcal{R}_i$ in $\mathtt{SERENA}$, this game excludes attacks by which $\mathcal{A}$ can win based on the condition (1).

Game $\mathbf{G}_3$. [*Aggregate signature forgery for $\bar{\sigma}_{1..i}$*] In this game the simulator chooses some index $i^* \in [1, n]$ at random. It then computes secret/public signing/verification keys (sk_i, vk_i) for each mesh router $\mathcal{R}_i$ with $i \neq i^*$ whereas for $\mathcal{R}_{i^*}$ it gets access to the aggregate signing oracle (representing the algorithm ASig). The simulation in this game is identical to Game $\mathbf{G}_2$ with the only exception that Δ fails if $\mathcal{A}$ asks a Send query containing a valid aggregate signature $\bar{\sigma}_{1..i^*}$ to an instance of $\mathcal{R}_{i^*+1}$ (or to an instance of $\mathcal{H}$ in case that $i^* = n$) such that $\bar{\sigma}_{1..i^*}$ has not been previously output by the ASig oracle on behalf of uncorrupted $\mathcal{R}_{i^*}$. The simulator can easily notice when such a Send query occurs.

In case that the simulation aborts Δ is in possession of a valid aggregate signature $\bar{\sigma}_{1..i^*}$ which was not obtained through any previous call to ASig, and can, therefore, be returned by Δ as a corresponding forgery. Hence,

$$|\Pr[\mathsf{Win}_3^{\text{ma-h-r}}] - \Pr[\mathsf{Win}_2^{\text{ma-h-r}}]| \leq n\mathsf{Succ}_{(\mathtt{ASig},\mathtt{AVer})}^{\mathtt{euf\text{-}cma}}(\kappa). \quad (6)$$

Since each $\bar{\sigma}_{1..i}$ is computed over the corresponding session id sid_i and the last aggregate signature $\bar{\sigma}_{1..n}$ is verified by $\mathcal{H}$ using own $sid_{\mathcal{H}}$ (amongst other inputs) and the session ids are unique according to the previous game, this game also rules out any successful replay attack for $\bar{\sigma}_{1..n}$. Since verification of $\bar{\sigma}_{1..n}$ by $\mathcal{H}$ is the necessary requirement for the acceptance of $\mathcal{H}$ in SERENA, this game excludes attacks by which $\mathcal{A}$ can win based on the condition (2).

Further, we focus on the attacks based on condition (3). Observe, that due to the inclusion of θ_p into the signed message the successful verification of $\sigma_{\mathcal{H}}$ by any mesh router $\mathcal{R}_i$ implies that $\mathcal{R}_i$ must have computed the same value for θ_p, and consequently both $\mathcal{R}_i$ and $\mathcal{H}$ hold identical k_p that they use to compute the path key K_p. Hence, the probability that partnered instances of some mesh router $\mathcal{R}_i$ and $\mathcal{H}$ accept with two different path keys in this game is 0, i.e. $\Pr[\mathsf{Win}_3^{\text{ma-h-r}}] = 0$. Combining the previous equations, we conclude the proof.

Theorem 4 (p-AKE). *Given EUF-CMA secure* (Sig, Ver) *and* (ASig, AVer)*, a IND-CCA2 secure* (Enc, Dec)*, and a pseudo-random* PRF *the protocol* SERENA *specified in Figure 1 provides p-AKE-security in the sense of Definition 7, and*

$$\mathsf{Adv}_{\mathtt{SERENA}}^{\text{e-ake}}(\kappa) \leq \frac{2(n+2)q^2}{2^\kappa} + 4\mathsf{Succ}_{\mathtt{MAC}}^{\mathtt{wuf\text{-}cma}}(\kappa) + 2\mathsf{Succ}_{(\mathtt{Sig},\mathtt{Ver})}^{\mathtt{euf\text{-}cma}}(\kappa) + \\ + 2n\mathsf{Succ}_{(\mathtt{ASig},\mathtt{AVer})}^{\mathtt{euf\text{-}cma}}(\kappa) + 2q\mathsf{Adv}_{(\mathtt{Enc},\mathtt{Dec})}^{\mathtt{ind\text{-}cca2}}(\kappa) + 8q\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa).$$

Proof. In the following we construct a sequence of games $\mathbf{G}_i$, $i = 0, \ldots, 7$ and denote by $\mathsf{Win}_i^{\text{p-ake}}$ the event that $\mathcal{A}$ breaks the p-AKE-security of SERENA in game $\mathbf{G}_i$, i.e., wins in the corresponding interaction as described in Definition 7.

Game $\mathbf{G}_0$. [*Real protocol*] This is the real $\mathsf{Game}_{\mathtt{SERENA}}^{\text{p-ake}}(\kappa)$ played between a simulator Δ and a PPT adversary $\mathcal{A}$. Δ simulates the actions of the participating $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ according to the protocol specification and answers all queries of $\mathcal{A}$. Recall, that the Test_Kp query is asked by $\mathcal{A}$ to a p-fresh instance of either $\mathcal{M}$, $\mathcal{H}$, or some $\mathcal{R}_i$ which has previously accepted. Note in particular, that the notion of p-freshness excludes any corruptions of $\mathcal{M}$, $\mathcal{H}$, and any $\mathcal{R}_i$. In order to prevent $\mathcal{A}$ from active participation on behalf of either of these parties we first exclude possible impersonation attacks.

Game $\mathbf{G}_1$. [*Collisions for nonces $r_{\mathcal{M}}$, $r_{\mathcal{H}}$, and $\{r_i\}_i$*] The simulation in this game aborts (and the output bit of the interaction is set at random) if the same random nonce is chosen by Δ on behalf of $\mathcal{F}$, $\mathcal{M}$, or $\mathcal{H}$, respectively, in two different protocol sessions. Thus,

$$|\Pr[\mathsf{Win}_1^{\text{p-ake}}] - \Pr[\mathsf{Win}_0^{\text{p-ake}}]| \leq \frac{(n+2)q^2}{2^\kappa}. \quad (7)$$

Game $\mathbf{G}_2$. [*MAC forgeries for $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$*] The simulation in this game aborts (and the output bit of the interaction is set at random) if $\mathcal{A}$ asks as part of its Send query to $\mathcal{M}$ (or to $\mathcal{H}$) a valid MAC value $\mu_{\mathcal{H}}$ (or $\mu_{\mathcal{M}}$) which was not previously output by an instance of $\mathcal{H}$ (or $\mathcal{M}$). Obviously,

$$|\Pr[\mathsf{Win}_2^{\text{p-ake}}] - \Pr[\mathsf{Win}_1^{\text{p-ake}}]| \leq 2\mathsf{Succ}_{\mathtt{MAC}}^{\mathtt{wuf\text{-}cma}}(\kappa). \quad (8)$$

The elimination of possible forgeries and replay attacks for $\mu_{\mathcal{H}}$ and $\mu_{\mathcal{M}}$ implies that any accepting p-fresh instance of $\mathcal{M}$ (or $\mathcal{H}$) has a partnered instance of $\mathcal{H}$ (or $\mathcal{M}$) which is also p-fresh.

Game $\mathbf{G}_3$. [*Signature forgery for $\sigma_{\mathcal{H}}$*] The simulation in this game aborts (and the output bit of the interaction is set at random) if $\mathcal{A}$ asks as part of its Send query to some $\mathcal{R}_i$ a valid signature $\sigma_{\mathcal{H}}$ which was not previously output by an instance of $\mathcal{H}$, so that

$$|\Pr[\mathsf{Win}_3^{\text{p-ake}}] - \Pr[\mathsf{Win}_2^{\text{p-ake}}]| \leq \mathsf{Succ}_{(\mathtt{Sig},\mathtt{Ver})}^{\mathtt{euf\text{-}cma}}(\kappa). \tag{9}$$

The elimination of possible forgeries and replay attacks for $\sigma_{\mathcal{H}}$ implies that any accepting p-fresh instance of a mesh router $\mathcal{R}_i$ has a partnered instance of $\mathcal{H}$ which is also p-fresh.

Game $\mathbf{G}_4$. [*Aggregate signature forgery for $\bar{\sigma}_{1..i}$*] The simulation in this game is identical to Game $\mathbf{G}_3$ with the only exception that Δ chooses an index $i^* \in [1, n]$ such that for the mesh router $\mathcal{R}_{i^*}$ it obtains aggregate signatures $\bar{\sigma}_{1..i^*}$ through an oracle and aborts the simulation if $\mathcal{A}$ asks a Send query containing a valid aggregate signature $\bar{\sigma}_{1..i^*}$ to an instance of $\mathcal{R}_{i^*+1}$ (or to an instance of $\mathcal{H}$ in case that $i^* = n$) such that $\bar{\sigma}_{1..i^*}$ has not been previously output by any call to the oracle.

In case that the simulation aborts Δ is in possession of a valid aggregate signature $\bar{\sigma}_{1..i^*}$ which was not obtained through any previous call to $\mathtt{ASig}$, and can, therefore, be returned by Δ as a corresponding forgery. Hence, $|\Pr[\mathsf{Win}_4^{\text{ma-h-r}}] - \Pr[\mathsf{Win}_3^{\text{ma-h-r}}]| \leq n\mathsf{Succ}_{(\mathtt{ASig},\mathtt{AVer})}^{\mathtt{euf\text{-}cma}}(\kappa)$.

The elimination of possible forgeries and replay attacks for any $\bar{\sigma}_{\{}1..i\}$ implies that any accepting p-fresh instance of $\mathcal{H}$ has a partnered instance of each $\mathcal{R}_i$ which is also p-fresh.

Games $\mathbf{G}_2$, $\mathbf{G}_3$, and $\mathbf{G}_4$ imply that if at least one p-fresh instance of some protocol party accepts then there exist partnered instances of all other parties which are also p-fresh.

Game $\mathbf{G}_5$. [*Pseudo-randomness of K_e*] This game is identical to Game $\mathbf{G}_4$ except that Δ in each session on behalf of any partnered instances of $\mathcal{M}$ and $\mathcal{H}$ chooses the end-to-end key K_e as a random value and not as an output of $\mathtt{PRF}$, s.t. $|\Pr[\mathsf{Win}_5^{\text{p-ake}}] - \Pr[\mathsf{Win}_4^{\text{p-ake}}]| \leq q\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa)$.

This game ensures independence between the key material k'_p that will be used to derive k_p, θ_p, and finally K_p, and the end-to-end key K_e which may be revealed by $\mathcal{A}$ without compromising the p-freshness of an instance.

Game $\mathbf{G}_6$. [*Pseudo-randomness of k'_p*] This game is identical to Game $\mathbf{G}_5$ except that Δ in each session on behalf of the partnered instances of $\mathcal{M}$ and $\mathcal{H}$ chooses k'_p as a random value and not as an output of $\mathtt{PRF}$, s.t. $|\Pr[\mathsf{Win}_6^{\text{p-ake}}] - \Pr[\mathsf{Win}_5^{\text{p-ake}}]| \leq q\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa)$.

Note that the chosen random value will also be used in the hop-by-hop encryption given by a sequence of $\chi_n, \ldots, \chi_1$.

Game $\mathbf{G}_7$. [*Security of χ_i*] In order to exclude any information leakage about k'_p upon its hop-by-hop encrypted transmission across the mesh network we consider the following game, in which Δ in each session computes each χ_i as $\mathtt{Enc}_{ek_i}(\beta)$ where β is some randomly chosen value, independent of k'_p. Note that Δ derives k_p, θ_p, and K_p on behalf of the partnered instances of $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ still using k'_p (and not β). It is possible to construct a distinguisher with given access to the *real-or-random* encryption oracle (and the decryption oracle) that is able to use $\mathcal{A}$ in this and the previous game to break the IND-CCA2 security of $(\mathtt{Enc}, \mathtt{Dec})$, s.t.

$$|\Pr[\mathsf{Win}_7^{\text{p-ake}}] - \Pr[\mathsf{Win}_6^{\text{p-ake}}]| \leq qn\mathsf{Adv}_{(\mathtt{Enc},\mathtt{Dec})}^{\mathtt{ind\text{-}cca2}}(\kappa). \tag{10}$$

Game $\mathbf{G}_8$. [*Pseudo-randomness of k_p and θ_p*] In this game Δ proceeds as before except that in each session on behalf of the partnered instances of $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ it chooses k_p and θ_p as two independent random values and not as an output of $\mathtt{PRF}$. Obviously,

$$|\Pr[\mathsf{Win}_8^{\text{p-ake}}] - \Pr[\mathsf{Win}_7^{\text{p-ake}}]| \leq q\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa). \tag{11}$$

Game $\mathbf{G}_9$. [*Pseudo-randomness of K_p*] In this final game Δ proceeds as before except that in each session on behalf of the partnered instances of $\mathcal{M}$, $\mathcal{H}$, and $\{\mathcal{R}_i\}_i$ it chooses the path key K_p at random and not as an output of $\mathtt{PRF}$. Obviously,

$$|\Pr[\mathsf{Win}_9^{\text{p-ake}}] - \Pr[\mathsf{Win}_8^{\text{p-ake}}]| \leq q\mathsf{Adv}_{\mathtt{PRF}}^{\mathtt{prf}}(\kappa). \quad (12)$$

As a result of this game the answer given to $\mathcal{A}$ in response to its $\mathsf{Test\_Kp}$ query to some p-fresh instance of $\mathcal{M}$, $\mathcal{H}$, or any $\mathcal{R}_i$ is a completely random value, regardless of the chosen bit b. Obviously, the probability of $\mathcal{A}$ to win in this game is given by the probability of a random guess, i.e. $\Pr[\mathsf{Win}_9^{\text{e-ake}}] = \frac{1}{2}$. The combination of the above equations concludes the proof.

5 Forward Secrecy, Anonymity, and Accounting

$\mathtt{SERENA}$ can be extended in a modular way to address forward secrecy, anonymity of mobile devices, and accounting between the cooperating networks of the same mesh.

Forward Secrecy of End-to-End and Path Keys The common way to achieve *forward secrecy*, i.e. to ensure that AKE-security holds even if $\mathcal{A}$ gains access to the long-lived keys of participants after their instances have accepted in the test session, is to derive the key from some ephemeral secret information which is valid only for one particular session.

Forward secrecy of K_e can be obtained using the classical Diffie-Hellman technique by deriving K_e from an ephemeral secret $g^{x_\mathcal{M} x_\mathcal{H}}$ (where $\mathbb{G}$ is a cyclic group of prime order q and $x_\mathcal{M}, x_\mathcal{H} \in \mathbb{Z}_q$), e.g. as an output of $\mathtt{PRF}_{f(g^{x_\mathcal{M} x_\mathcal{H}})}(l_3|sid)$ with some randomness extractor f [12]. For this, $\mathcal{M}$ and $\mathcal{H}$ must exchange $g^{x_\mathcal{M}}$ and $g^{x_\mathcal{H}}$ as part of their first protocol messages. In order to ensure authentication, $\mathcal{H}$ and $\mathcal{M}$ must include $g^{x_\mathcal{M}}$ and $g^{x_\mathcal{H}}$ into the computation of $\mu_\mathcal{H}$ and $\mu_\mathcal{M}$, respectively. The AKE-security of such K_e would then rely on the Decisional Diffie-Hellman Problem.

Forward secrecy of K_p can be obtained similarly using the Generalized Diffie-Hellman technique [8] by deriving K_p from $g^{x_\mathcal{M}(\prod_i x_i)x_\mathcal{H}}$ (where $x_\mathcal{M}, x_i, x_\mathcal{H} \in \mathbb{Z}_q$ and each x_i is chosen by $\mathcal{R}_i$). Note that in this case there is no need to apply hop-by-hop encryption of k'_p so that all computations involving k'_p become obsolete and the key pairs (dk_i, ek_i) can be dismissed from the routers' long-lived keys. The AKE-security of K_p would then rely on the Group Decisional Diffie-Hellman Problem. Note that achieving forward secrecy increases the protocol costs and if forward secrecy is required for K_e and K_p simultaneously then exponents $x_\mathcal{M}$ and $x_\mathcal{H}$ should be independent for each of the above techniques.

Anonymity of $\mathcal{M}$ If necessary $\mathtt{SERENA}$ can be extended to achieve anonymity of $\mathcal{M}$ and unlinkability of its sessions simply by encrypting the identity $\mathcal{M}$ in the first protocol message using the IND-CCA2 secure encryption scheme $(\mathcal{E}, \mathcal{D})$. Here we assume that $\mathcal{H}$ holds own decryption/encryption key pair $(dk_\mathcal{H}, ek_\mathcal{H})$ as part of its long-lived key, which is already the case as $\mathcal{H}$ is one of the networks of the mesh. This implies that the session id would be computed using the encryption of $\mathcal{M}$. We remark that this solution increases the costs by one public key operation for both $\mathcal{M}$ and $\mathcal{H}$.

Accounting between $\mathcal{H}$ and $\mathcal{R}_i$ In $\mathtt{SERENA}$ each mesh router obtains $\sigma_\mathcal{H}$ computed by $\mathcal{H}$ amongst other parameters on the session id which also includes the identity of each $\mathcal{R}_i$. This signature can be extended with the time-stamp T and used for the purpose of accounting between $\mathcal{H}$ and $\mathcal{R}_i$, e.g. as an incentive mechanism for the cooperation of mesh routers. Note that since the packets are routed in real-time, both $\mathcal{H}$ and each $\mathcal{R}_i$ can independently keep track on the size of the transmitted packets or on the duration of the session. Further, any signature $\sigma'_\mathcal{H}$ computed by $\mathcal{H}$ for some time interval $[T, T']$ could serve as a cryptographically protected acknowledgement of $\mathcal{H}$ for the reimbursement of $\mathcal{R}_i$.

6 Conclusion

The relatively stable routing infrastructure provided by wireless mesh networks offers valuable connectivity service to the mobile clients. In this paper we addressed the remote wireless network access of a mobile client to its home network inside the same mesh. We refined the classical end-to-end security approach by the additional concept of path security and argued on its benefits with regard to security and robustness of the remote session. We formalized new goals using well-known and recognized model for authentication and key establishment and proposed a provably secure, yet lightweight, round-optimal, modular and thus extensible, remote network access protocol `SERENA` that binds the end-to-end protected application channel to the underlying stable routing path. We also mentioned how `SERENA` can be effectively realized via combination of well-known authentication standards IEEE 802.1X, EAP, PANA, and IPsec.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *CT-RSA'01*, *LNCS* 2020, pp. 143–158. Springer, 2001.
2. B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748, IETF, 2004.
3. M. Bellare, R. Canetti, and H. Krawczyk. Keying Hash Functions for Message Authentication. In *CRYPTO '96*, *LNCS* 1109, pp. 1–15. Springer, 1996.
4. M. Bellare and C. Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In *ASIACRYPT'00*, *LNCS* 1976, pp. 531–545. Springer, 2000.
5. M. Bellare and P. Rogaway. Entity Authentication and Key Distribution. In *CRYPTO '93*, *LNCS* 773, pp. 232–249. Springer, 1993.
6. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In *EUROCRYPT'96*, *LNCS* 1070, pp. 399–416. Springer, 1996.
7. F. Bersani and H. Tschofenig. The EAP-PSK Protocol: A Pre-Shared Key EAP Method. RFC 4764, IETF, 2007.
8. E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. In *CCS'01*, pp. 255–264. ACM, 2001.
9. L. Buttyán and J.-P. Hubaux. *Security and Cooperation in Wireless Networks*. Cambridge Univ. Press, 2008.
10. O. Cheikhrouhou, M. Laurent-Maknavicius, and H. Chaouchi. Security Architecture in a Multi-Hop Mesh Network. In *SAR'06*, 2006.
11. T. Clancy and W. Arbaugh. EAP Password Authenticated Exchange. RFC 4746, IETF, 2006.
12. Y. Dodis, R. Gennaro, J. Håstad, H. Krawczyk, and T. Rabin. Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes. In *CRYPTO'04*, *LNCS* 3152, pp. 494–510. Springer, 2004.
13. R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. In *SIGCOMM'04*, pp. 133–144. ACM, 2004.
14. D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin. Protocol for Carrying Authentication for Network Access (PANA). RFC 5191, IETF, 2008.
15. P.-A. Fouque, D. Pointcheval, and S. Zimmer. HMAC is a Randomness Extractor and Applications to TLS. In *ACISP'03*, *LNCS* 2727, pp. 180–191.
16. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP Is Secure under the RSA Assumption. *Journal of Cryptology*, 17(2):81–104, 2004.
17. P. Funk and S. Blake-Wilson. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). RFC 5281, IETF, 2008.
18. D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728, IETF, 2007.
19. K. Khan and M. Akbar. Authentication in Multi-Hop Wireless Mesh Networks. *PWASET*, 16:178–183, 2006.
20. S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters. Sequential Aggregate Signatures and Multisignatures without Random Oracles. In *EUROCRYPT'06*, *LNCS* 4004, pp. 465–485. Springer, 2006.

21. A. Lysyanskaya, S. Micali, L. Reyzin, and H. Shacham. Sequential Aggregate Signatures from Trapdoor Permutations. *EUROCRYPT'04*, *LNCS* 3027, pp. 74–90. Springer, 2004.
22. H. Moustafa, G. Bourdon, and Y. Gourhant. Authentication, Authorization and Accounting (AAA) in Hybrid Ad hoc Hotspot's Environments. In *WMASH'06*, pp. 37–46. ACM, 2006.
23. NIST. Digital Signature Standard (DSS). FIPS PUB 186-2, 2000.
24. C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, IETF, 2003.
25. D. Simon, B. Aboba, and R. Hurst. The EAP-TLS Authentication Protocol. RFC 5216, IETF, 2008.
26. S. Xu, Y. Mu, and W. Susilo. Online/Offline Signatures and Multisignatures for AODV and DSR Routing Security. In *ACISP'06*, *LNCS* 4058, pp. 99–110. Springer, 2006.
27. M. Zhao, S. W. Smith, and D. M. Nicol. Aggregated Path Authentication for Efficient BGP Security. In *ACM CCS'05*, pp. 128–138. ACM, 2005.
28. H. Zhu, F. Bao, T. Li, Y. Wu. Sequential Aggregate Signatures for Wireless Routing Protocols. In *IEEE WCNC'05*, pp. 2436–2439. IEEE, 2005.
29. S. Zhu, S. Xu, S. Setia, S. Jajodia. LHAP: A Lightweight Network Access Control Protocol for Ad Hoc Networks. *Ad Hoc Networks*, 4(5): 567-585. Elsevier, 2006.