Tiek: Two-tier Authentication and Key Distribution for Wearable Devices

Sam Hylamia
Uppsala University
Uppsala, Sweden
sam.hylamia@it.uu.se

Wenqing Yan
Uppsala University
Uppsala, Sweden
wenqing.yan@it.uu.se

Christian Rohner
Uppsala University
Uppsala, Sweden
christian.rohner@it.uu.se

Thiemo Voigt
Uppsala University & RISE
Uppsala & Stockholm, Sweden
thiemo.voigt@it.uu.se

***Abstract*—Wearable devices, such as implantable medical devices and smart wearables, are becoming increasingly popular with applications that vary from casual activity monitoring to critical medical uses. Unsurprisingly, numerous security vulnerabilities have been found in this class of devices. Yet, research on physical measurement-based authentication and key distribution assumes that body-worn devices are benign and uncompromised. Tiek is a novel authentication and key distribution protocol which addresses this issue. We utilize two sources of randomness to perform device authentication and key distribution simultaneously but through separate means. This creates a two-tier authorization scheme that enables devices to join the network while protecting them from each other. We describe Tiek and analyze its security.**

I. Introduction

Wearable devices span various application areas from health care to information delivery and fitness tracking. Smart wearables often measure, communicate and store sensitive and personal information, such as user identification, physiological signals, location data, etc. Some even provide life-saving and critical functions, like cardiac defibrillators and insulin pumps. Hence, securing these devices is imperative [**?**].

Authentication and key distribution/agreement are fundamental challenges in communication security. The unique environment available to wearable devices enables a novel technique that addresses these challenges, namely physical and physiological measurement-based authentication and key agreement [**?**], [**?**], [**?**], [**?**], [**?**]. Systems employing this technique provide an alternative to traditional methods that rely on computational hardness and available infrastructure, such as Public Key Infrastructure (PKI). This paper leverages this technique and presents a novel authentication and key distribution protocol named Tiek. We enhance state-of-the-art systems by fortifying distributed keys against mal-behaving body-worn devices.

Physical and physiological measurement-based authentication and key agreement techniques utilize their environment to generate cryptographic keys and authenticate devices without any prior knowledge. These techniques typically rely on a physical time-varying random process available to legitimate participating devices. An eligible random process produces correlated results when measured independently by different devices at the same time. Examples of such processes are the inter-pulse interval (IPI) of a heart [**?**], body movement patterns [**?**] and wireless channel fading [**?**]. Participating devices obtain correlated random strings by simultaneously measuring one of these processes. They, then, use these strings to authenticate each other [**?**] or derive keys [**?**]. We refer to these random processes as sources of randomness and divide them into two categories:

This project is financed by the Swedish Foundation for Strategic Research.

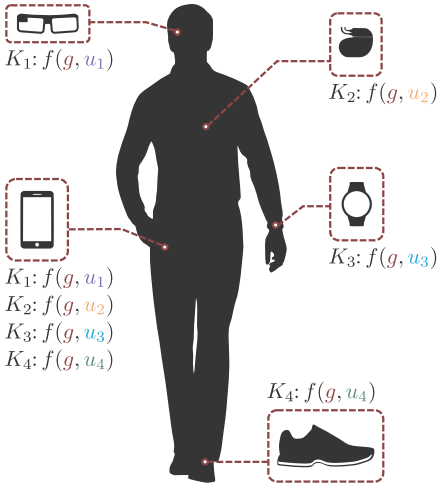

Fig. 1. A smartphone and four other smart devices use Tiek to establish secure communication. Each device obtains the group portion g and its unique portion u_i and fuses them to create the final key K_i.

1) *Common sources*, such as IPI and body movement patterns, are physiological features that can be measured accurately enough anywhere on the body. When measured independently and at the same time, these sources provide correlated values to all devices sharing the body.

2) *Pairwise sources*, such as wireless channel fading, are only available to a pair of collaborating devices. When measured independently and simultaneously by Alice, Bob and Eve, a pairwise source provides different values to Alice and Bob than Alice and Eve or Bob and Eve.

To the extent of our knowledge, all systems that employ the aforementioned technique assume that participating body-worn devices are benign and untampered with [**?**], [**?**]. However, studies have uncovered numerous security vulnerabilities in wearable devices [**?**], [**?**]. Such vulnerabilities could compromise a body-worn device enabling an adversary to gain substantial knowledge of the secrets of the compromised de-

vice; therefore, jeopardizing the security of such systems. We challenge this commonly accepted assumption and consider the situations where a body-worn device exhibits malicious behaviour enabled, for example, by an exploited vulnerability.

The Tiek protocol. On a high level, Tiek is a multilevel authentication and key distribution protocol for wearable devices. In essence, our protocol utilizes a common source of randomness and a pairwise source to *distribute* a unique key to every device in the network. We generate all keys at a relatively powerful hub device and use the sources of randomness to distribute them securely.

Each key consists of two parts that we fuse together into the final key. The first part of the key is distributed through the common source and is known to all devices sharing the body. For example, all devices in Figure 1 measure the body's motion pattern (gait) and, through that, decode the same bit-string g. We call g the *group portion* and use it to grant (authorize) devices access to the network. In other words, devices authenticate their presence on the body by demonstrating their knowledge of the group portion to the hub. The second part is called the *unique portion* and is distributed through the pairwise source. This portion is shared between every individual device and the hub. In our previous example, all devices in Figure 1 measure the channel fading between themselves and the smartphone and use their measurements to decode the unique portion u_i. We use the unique portion to tie the final key to a specific device. For the *final key*, we fuse the group portion and the unique one to obtain a cryptographic key unique to every device in the network. All final keys are known to the hub where they are used to establish an authenticated and encrypted channel with every device. The final key is denoted K_i in Figure 1 where f is the fuse portions function.

Our method leverages this two-part key to establish two levels of authorization in the network. The first level is granted by the group portion and constitutes the ability to participate in the network. Therefore, any body-worn device, including mal-behaving ones, can communicate with the hub and appear as authentic devices. For example, all devices in Figure 1 know the group portion g and are, therefore, allowed to communicate with the smartphone. However, this, unfortunately, implies that if an off-body adversary learns g through a vulnerability in one of the devices, it will be able to communicate with the smartphone. The second level is where Tiek mitigates the risk mal-behaving body-worn devices pose to the network. Since every key is constructed with a unique portion, only devices that share this portion are authorized to communicate with each other. Hence, in our previous example, only the smartphone is allowed to communicate with the implantable medical device since only those two know u_2.

As a result of this scheme, Tiek utilizes the security of the full final key against off-body adversaries. Simultaneously, it utilizes the significant but potentially reduced security of the unique portion against body-worn devices courtesy of their knowledge of the group portion. Additionally, since Tiek employs two independent sources of randomness it can alleviate the potential damage in situations where one of the sources is compromised.

Contributions.

- We leverage the variety of randomness sources available to wearable devices to design the first protocol to utilize different sources of randomness for authentication and key distribution for wearable devices.
- Our protocol is also the first design to mitigate the threats a mal-behaving body-worn device poses to the secrecy of the distributed keys.

Organization. The rest of this paper is organized as follows: §II provides a summary of the related work and how it compares to ours. In §III, we state our goals, assumptions and adversary model. Then, we describe Tiek in §IV and analyze its security in §V. Finally, we conclude this work in §VI.

II. Related Work

Various studies have adopted physical and physiological measurements-based authentication and key agreement as an alternative to those based on computational hardness. In this section, we categorize existing studies based on their utilized source of randomness and compare Tiek with the state-of-art.

Common source based schemes. Several studies have proposed to use biometric and physiological signals to authenticate devices and agree on keys, such as IPI [**?**], [**?**], hand resonance [**?**] and gait [**?**]. Revadigar et al. [**?**] utilize human motion to generate and distribute a group key between multiple wirelessly connected wearable devices. The proposed scheme uses an accelerometer to generate a random bit-string (group key) at a hub device. Then, all devices extract gait and use it to distribute the group key though a fuzzy vault scheme. This method of key distribution assumes that all body-worn devices are benign since a malicious wearable could bypass the protection offered by this scheme. In contrast, Tiek mitigates the potential of a malicious wearable device by complementing the group key with another one unique to every device. We obtain the unique key portions from a pairwise source which enables us to protect devices from each other.

Pairwise source based schemes. Existing schemes using pairwise sources are concentrated around wireless channel characteristics, such as channel fading [**?**] and channel anonymity [**?**]. Shi et al. [**?**] use received signal strength (RSS) to authenticate wearable devices and generate a unique key between each wearable and a hub. The proposed scheme leverages the significant difference in RSS fluctuations between two body-worn devices and between a body-worn device and an off-body device. This difference is used to prove a device's presence on the body. Then, devices whose presence on the body has been authenticated collaborate to establish a unique key between each one of them and the hub. Similar to Revadigar et al., Shi et al. do not consider malicious body-worn devices. Moreover, their method requires additional wearable devices to facilitate their authentication scheme. We differentiate from their approach by separating the means of device authentication from those of key distribution. This enables us to establish clear boundaries between body-worn devices and protect them from

each other. Moreover, our method does not require additional devices to facilitate its operations.

III. System Model

A. Design Goals and Requirements

Our main design objective is to create an authentication and key distribution protocol that alleviates the risk of a system-wide security failure against mal-behaving body-worn devices. The protocol must be realizable using existing hardware and standardized protocols. It must also fit energy, memory and time constraints of wearable devices by minimizing the communication and computation overhead. Moreover, the system must not compromise usability by minimizing user interaction.

B. Assumptions

We consider a network of wirelessly connected wearable devices, such as the one shown in Figure 1. We assume that all devices can measure a common source and a pairwise source of randomness. For example, all devices can measure gait using an on-board accelerometer, and RSS using their radio. A dedicated device in the network, e.g. the smart phone, acts as a communication hub between body-worn devices and between body-worn devices and the outside world. In other words, all devices form a star topology network around the hub. We refer to the hub as the central control unit (CU).

C. Adversary Model

In this work, we identify two types of adversaries:

1) *Off-body adversaries* are the typical type of adversaries discussed in previous systems [**?**], [**?**]. We define off-body adversaries as passive and active attackers that cannot measure the utilized sources of randomness accurately. In passive attack scenarios, off-body adversaries can eavesdrop on all exchanged messages between legitimate devices. Eavesdroppers analyze these messages to learn about the distributed keys. In active attack scenarios, off-body adversaries attempt to impersonate a body-worn device. Here, adversaries can employ advanced sensing techniques to learn about the distributed keys, such as measuring IPI using cameras [**?**] or imitating a person's gait [**?**]. The goal of an off-body adversary is to gain access to the network to eavesdrop on confidential communication or launch subsequent attacks on other devices.

2) *Body-worn adversaries* are body-worn devices that exhibit malicious behaviour caused by a vulnerability. Such an adversary can measure the common source of randomness accurately. This effectively grants the body-worn adversary access to the network. In other words, this adversary can establish an authenticated and confidential channel with the CU. The goal of a body-worn adversary is to eavesdrop on the confidential communication between the CU and other body-worn devices or attack a benign body-worn device. As mentioned in the introduction, previous systems have disregarded body-worn adversaries. Therefore, considering them in our adversary model and limiting their potential in our design are among our contributions.

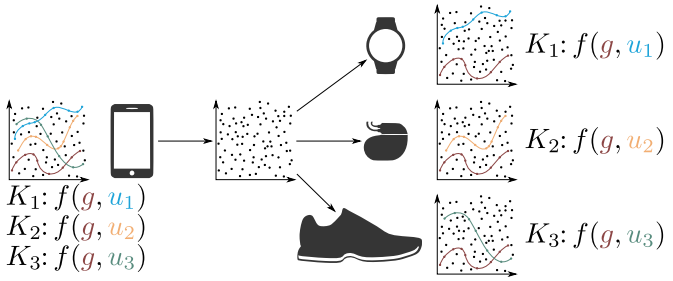

Fig. 2. Illustration of the key distribution technique utilized in Tiek. Each key is encoded as points on a polynomial in the fuzzy vault and obfuscated by a set of chaff points. Each node is able to decode only its unique portion u_i and the group portion g from the vault.

We assume that the CU is benign and untampered with. This is an exception to our previous assumptions because of the central role of this device in our protocol and the network as a whole. Similar to previous work [**?**], [**?**], we do not consider jamming and denial-of-service (DoS) attacks. We assume that the CU can detect jamming and DoS attacks, interrupt the key agreement procedure and alert the user.

IV. The Tiek Protocol

As introduced earlier, Tiek utilizes a common source of randomness and a pairwise source to authenticate devices and distribute keys. Tiek, in essence, is a *key distribution* scheme where we generate keys independently from the utilized sources of randomness but distribute them through these sources. The proposed method is orchestrated by the CU and follows an intentionally simple, yet effective, procedure.

As shown in Figure 1, we assume a network of a CU and N wearable devices, denoted $\{n_i\}$ for $i \in [1, N]$. We refer to all devices other than the CU as nodes. In this network, Tiek generates and distributes keys following a three-step process: First, the CU generates all key portions needed to construct the final keys. Next, all devices utilize the sources of randomness to distribute each generated key portion to its target node. Finally, each node combines its key portions and obtains its final key. We detail these steps below.

A. Secret Key Material (SKM) Generation

Leveraging the superior resources available to the CU compared to other devices, the CU generates $N + 1$ true random and unique strings. These strings are the *group portion* g and the N *unique portions* $\{u_i\}$. The SKM can be obtained from a true physical random number generator on-board the CU or communicated securely from a trusted third party.

B. Key Distribution

Following SKM generation, the CU needs to distribute these strings to all participating nodes securely without any prior shared keys or infrastructure. Hence, Tiek utilizes a cryptographic construct known as a multi-secret fuzzy vault [**?**]. This scheme enables us to distribute all key portions (SKM) to all devices simultaneously without leaking any information. We first introduce the fuzzy vault scheme and then extend to its multi-secret variant. *Fuzzy vault* is an information-theoretically

secure construct proposed by Juels and Sudan [**?**]. It enables us to distribute a secret though a "vault" using two operations, namely, vault locking and unlocking. To lock a vault, we encode the secret using a set of random values (vault key) and add a set of noise (chaff) to obfuscate the encoded secret. Then, we publish the vault which contains our encoded secret and the noise. Devices receiving the vault can unlock it and retrieve the secret by demonstrating substantial knowledge of the vault key. To unlock the vault, a device must use its version of the vault key to pick out a substantial number of correct values from the vault and decode the secret. *Multi-secret fuzzy vault* [**?**] is an extension to the fuzzy vault scheme. It allows us to encode and decode multiple secrets using multiple keys in the same vault. In Tiek, the vault keys are obtained from the sources of randomness and encode the generated SKM.

The key distribution protocol as illustrated in Figure 2 is detailed as follows:

1) *Vault Key Extraction:* At this step, we extract the sets of values (vault keys) we use to encode the SKM and construct the vault. These sets are sampled from the common and pairwise sources of randomness as follows: (a) The CU initiates the vault key extraction procedure and provides the necessary parameters devices need to collect the right set of random values. (b) Upon receiving the initiation signal, all devices, including the CU, sample the common source and retrieve an unordered set of random values (common set). This set is the vault key that encodes/decodes the group portion g. (c) During the common source sampling procedure, the CU collaborates with the nodes individually to sample the pairwise source and retrieve an unordered set of values (pairwise set) for every node. These sets are the vault keys that encode/decode the unique portions $\{u_i\}$. For details on how these values are sampled and processed we refer to earlier work [**?**], [**?**], [**?**], [**?**].

The result of this procedure at each node is a set of values correlated between all devices and another set correlated between the CU and the node itself. At the CU, these vault keys are denoted A_0 for the common set and $\{A_i\}$ for the pairwise sets. The counterpart of these sets at the nodes are denoted $\{B_j\}$ for $j \in [0, N]$ respectively.

2) *Vault Locking:* The goal of this step is to construct and publish a multi-secret vault that enables nodes to retrieve their SKM. Using the vault keys extracted in the previous step, the CU constructs a vault containing all SKM and publishes it to participating devices. The vault construction process proceeds as follows: (a) The CU creates a polynomial p_j for every SKM by breaking the SKM into small consecutive blocks of length s and using them as the polynomial coefficients. These polynomials are p_0 for SKM g and $\{p_i\}$ for SKM $\{u_i\}$. The order of polynomial p_j is denoted k_j and equals the length of the SKM it encodes divided by the block size s minus one. (b) Following polynomial construction, the CU projects each vault key A_j on its respective polynomial p_j and obtains the sets of points $\left\{P_j = \left(A_j^l, p_j\left(A_j^l\right)\right) | l \in [0, \text{len}\left(A_j\right)[\right\}$. (c) Additionally, we create a set of random chaff points C. Elements in C are not members of any set $\{A_j\}$ and do not intersect with any polynomial $\{p_j\}$. Chaff points are generated randomly and must not be distinguishable from points in $\{P_j\}$ [**?**]. (d) All generated points are combined to create the set $V = P_0 \cup ... \cup P_N \cup C$. Set V is our vault which contains all the points required to retrieve all SKM, in addition to chaff points designed to throw an attacker off any useful information. Figure 2 illustrates a multi-secret fuzzy vault containing four secrets. Any node n_i that receives the vault will be able to retrieve only SKM g and u_i through demonstrating their substantial knowledge of the vault keys A_0 and A_i. To this node, the rest of the points in V will appear as part of the chaff. (e) Finally, the CU broadcasts the vault to all participating devices.

3) *Vault Unlocking*: At this step, all nodes have the vault and the keys required to unlock their key portions. The goal here is for every node n_i to extract the group portion g and its unique portion u_i from the vault. This step proceeds as follows: (a) Node n_i extracts two sets of points from the vault V, namely set R_0 and R_i. Set R_0 is selected such that its elements are in set B_0. Similarly, set R_i elements are in set B_i. (b) By applying Lagrange interpolation to the sets R_0 and R_i, each node n_i reconstructs the polynomials p_0 and p_i. (c) Node n_i then retrieves the SKM g and u_i from the coefficients of the reconstructed polynomials; hence receiving the group and the unique portions of its key.

C. Key Combination

At this point, every node has the group portion and unique portion of its key. These portions can be fused into the final key in various configurations influenced by the required security and the available resources. We use a key derivation function (KDF) to securely combine the two portions into the final key.

HKDF [**?**] is a KDF that relies on HMAC in a two-phase operation known as extract-then-extend. In the extract phase, a randomness extractor (XTR) samples the input SKM and produces a close-to-random output. The output of the XTR is then fed to a variable-length pseudorandom function (PRF*) to extend it to the required length or to generate multiple keys. In HKDF both XTR and PRF* are implemented using HMAC with an optional salt as a key in the extract phase. We use HKDF because of its general applicability and simplicity.

We slightly alter the extract phase of HKDF to fit our purpose. Instead of using the *extract phase* to extract randomness from an SKM, we utilize it to fuse the two key portions securely. We run HMAC on the unique portion u_i and replace the salt with the group portion g; $K_i = \text{HMAC}\left(g, u_i\right)$. In the *extend phase*, we derive an x number of session keys from K_i which we use to enforce confidentiality and authenticity of exchanged messages between the node and the CU. Each session key is used for some hours then replaced by the next session key. When all x session keys have expired, the Tiek protocol restarts and replaces the master key.

In some situations, the cost of vault key extraction is prohibitive of distributing sufficiently long SKM. This can be rectified by using a key stretching primitive, such as Argon2 [**?**], as the XTR of HKDF. Note that this only increases the resources needed to brute-force the key, however, does not

introduce any additional entropy. We maximize the length of the SKM before applying any key stretching technique.

The master keys $\{K_i\}$ Tiek generates are composed of both the group and unique portions; thus enforcing the knowledge of both. An adversary must access the common source to participate in the network and must brute-force the unique key portion of every node it intends to attack individually. This makes Tiek resilient to off-body and body-worn adversaries.

V. Security Analysis

In this section, we analyze how Tiek fairs against the adversaries we consider in the adversary model. First, we analyze the security of key distribution through the multi-secret fuzzy vault scheme. Then, we elaborate on the properties of the distributed keys. Finally, we discuss different attack scenarios.

Throughout our analysis, we consider an example of four wearable devices ($N = 4$) and a CU, such as that in Figure 1. The network uses Tiek to agree on an individual key between each device and the CU.

A. Security of the Vault

The security offered by the fuzzy vault depends on the order of the encoded polynomial and the number of chaff points added. The minimum size of a vault key set that encodes/decodes a polynomial of order k is $k+1$. In other words, for node n_i to reconstruct the polynomial p_j of order k_j, R_j must contain $k_j + 1$ correct points; $\text{len}\,(R_j \cap P_j) \geq k_j + 1$. Recall that P_j is the projection of the vault key A_j on p_j, and R_j is the set of vault points whose elements are in B_j.

During the key distribution phase of Tiek in our analysis example, the CU publishes a vault V containing five polynomials which represent the group portion g and four unique portions $\{u_i\}$. The vault also contains 200 chaff points ($\text{len}\,(C) = 200$). To simplify, we assume that all polynomials are of the same order $k = 19$. Hence, the size of our vault $\text{len}\,(V) = 300$ points. We use this example to discuss two scenarios where a passive adversary (eavesdropper) attempts to leak information about one or more keys from the vault.

In the first scenario, we consider a passive off-body adversary attempting to eavesdrop on the communication between the CU and a body-worn device. The adversary receives the vault and attempts to brute-force the key portions encoded in the vault. To violate the secure communication between the CU and a node n_i, this adversary needs to brute-force the group portion g and the node's unique portion u_i. Suppose the adversary tries every combination of 20 out of the 300 points. This sums up to over $7.5e30$ possible combinations which equals the requirements of brute-forcing a 102.56-bit key; $\log_2 \binom{300}{20} \approx 102.56$.

For the second scenario, we assume a body-worn adversary as defined in our adversary model. This adversary is already part of the network and knows the group portion as well as its unique portion. This leaves our vault with three encoded polynomials unknown to this adversary. In other words, if we assume that node n_1 is the body-worn adversary, then g and u_1 are revealed while u_2, u_3 and u_4 are still secret. Hence, the vault size is reduced to 260 points. Following our previous calculations, a body-worn adversary tries $3.87e29$ different combinations which equals a 98.28-bit key.

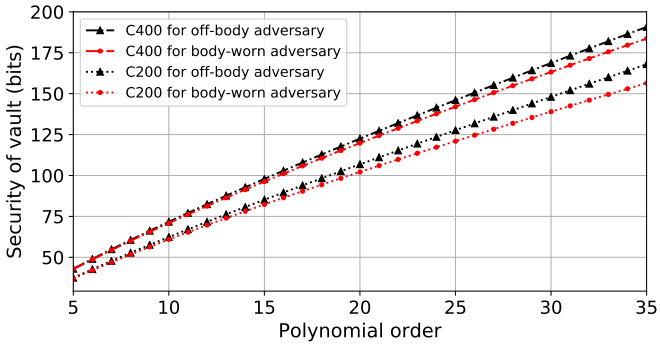

Fig. 3. Security of the vault in relation to the polynomial order and the chaff size for four devices. The security of the vault increases with higher polynomial orders and a larger chaff. A body-worn adversary only slightly reduces the security of the vault. Note that C400 and C200 refers to a chaff size of 400 and 200 points respectively.

In both scenarios, we find that Tiek achieves more than the recommended minimum of 85-bit security [**?**] even with a small 300-point vault. Figure 3 shows the security of the vault in relation to the polynomial order and the vault size in both scenarios. The security of the vault in our example can be improved by choosing a higher polynomial order or adding more chaff points.

There is an indirect relationship between the polynomial order and the cost of key distribution. Generally, higher polynomial orders require longer vault keys and, by conjunction, more measurements of the randomness sources. Previous work has shown that group sources of randomness can provide more bits per second compared to pairwise sources [**?**], [**?**], [**?**]. In Tiek this speed disparity gives system designers a trade-off between security and cost: On one hand, it is cheaper to increase security against off-body adversaries by using a polynomial of higher order to encode the group portion rather than the unique one. On the other hand, encoding the unique portion with a polynomial of higher order increases security against body-worn adversaries, as well as off-body ones, but comes at a higher cost. Moreover, different unique portions can use different polynomial orders which enables designers to further adjust Tiek to each device's security needs and available resources.

B. Properties of the Distributed Keys

As introduced, the master keys $\{K_i\}$ are composed of two portions. We elaborate on their properties and use:

Group portion g is Tiek's way of authenticating a device's presence on the body. In other words, if a device can retrieve g from the vault, it is considered an authentic device which authorizes it to participate in the network. Another possible use of g is to broadcast confidential messages to all body-worn devices. Broadcast messages can be encrypted using a key derived from the group portion alone. The authentication and broadcast security are only as strong as g.

Unique portions $\{u_i\}$ are the way Tiek enforces the second level of authorization and enables the CU to establish secure communication with individual devices. In other words, only node n_i can retrieve u_i from the vault which makes it the only node capable of constructing K_i. Therefore, K_i is as strong as the combination of g and u_i against off-body adversaries, but only as strong as u_i against body-worn ones.

The length of a key portion is a function of the polynomial order and the block size. Thus, by increasing the block size we can increase the length of a key portion at no additional cost. However, in reality, this may not contribute anything to the security of the system, because it is only as secure as the weakest point. For example, if we distribute 256-bit key portions using our vault from earlier, we gain no additional security over that provided by the vault. Simply because the adversary will resort to breaking the vault instead of the harder key portions. Therefore, we choose the key portion to match the security of the vault.

C. Attack Scenarios

We assume the presence of two adversaries in our analysis example: An off-body adversary within eavesdropping distance and one of our four wearables as a body-worn adversary. In the following, we identify multiple active attack scenarios and discuss them in light of our adversary model.

Adversary joining the network: The goal of the adversary is to impersonate a body-worn device and join the network. To join the network, an adversary would need to possess the group portion and its unique portion. While obtaining a unique portion from the CU is as easy as asking for it, the group portion bounds the presence of the device to the body. In other words, body-worn devices, including mal-behaving ones, can join the network, whereas off-body adversaries cannot obtain all the necessary key parts to participate. However, this property is violated on occasions when an off-body adversary obtains substantial information on the group portion through a side channel, such as advanced remote sensors or a vulnerable body-worn device. For instance, if a body-worn device has a vulnerability that reveals the group portion, an off-body adversary could exploit it to join the network. Note that an adversary that joins the network cannot eavesdrop on the communication between the CU and any other node without breaking the unique portion of that node. We can set a device count limit to mitigate DoS attacks where an adversary exploits this vulnerability to add many devices to the network.

Adversary impersonating the CU: This attack scenario is similar to the previous one, except, here, the adversary attempts to join the network as the CU. This is especially dangerous because if an adversary succeeds, it could control the entire network. To carry out this attack, an adversary must be able to measure both sources of randomness to encode its own key portions such that legitimate nodes are able to retrieve them. In other words, a body-worn adversary can easily impersonate the CU and publish its own vault. Hence, we propose an additional step in the protocol to allow nodes to verify the identity of the CU: Following the key combination step in our protocol, the CU commits to a random value v_i for each node n_i and binds it to K_i. The next time Tiek distributes keys, the CU decommits v_i, commits to a new one and binds it to the newly distributed key. Assuming that a legitimate CU distributes keys using Tiek the first time, this commitment scheme ensures that only the same CU can run Tiek successfully the next times.

Side-channel attacks on the sources of randomness: These attacks are tied to the specific sources we utilize. In general, all the security requirements of the sources must hold for Tiek to function at its highest level of security. However, as discussed earlier, we design Tiek to tolerate situations where an adversary obtains unauthorized access to (compromises) a source of randomness through a side channel. Tiek only fails at the severest scenarios where both sources of randomness are compromised. However, successful side-channel attacks on the sources of randomness (even those that reveal insubstantial information) can reduce the overall provided security.

VI. Conclusions

We provide a detailed description and security analysis of a novel authentication and key distribution protocol for wearable devices. We improve state-of-the-art by considering body-worn adversaries and mitigating the risk they pose. To this end, we utilize a common source of randomness to authenticate the presence of the devices on the body and leverage a pairwise source to tie each final key to a specific device. Finally, we show that our protocol is immune against eavesdropping and impersonation attacks under our adversarial model.