

Secure and Private, yet Lightweight, Authentication for the IoT via PUF and CBKA

Christopher Huth^{1(✉)}, Aydin Aysu², Jorge Guajardo³, Paul Duplys¹,
and Tim Güneysu⁴

¹ Robert Bosch GmbH, Stuttgart, Germany
`christopher.huth@de.bosch.com`

² The University of Texas at Austin, Austin, USA

³ Robert Bosch LLC - Research and Technology Center, Pittsburgh, USA

⁴ University of Bremen & DFKI, Bremen, Germany

Abstract. The Internet of Things (IoT) is boon and bane. It offers great potential for new business models and ecosystems, but raises major security and privacy concerns. Because many IoT systems collect, process, and store personal data, a secure *and* privacy-preserving identity management is of utmost significance. Yet, strong resource limitations of IoT devices render resource-hungry public-key cryptography infeasible. Additionally, the security model of IoT enforces solutions to work under memory-leakage attacks. Existing constructions address either the privacy issue or the lightwightness, but not both. Our work contributes towards bridging this gap by combining physically unclonable functions (PUFs) and channel-based key agreement (CBKA): (i) We show a flaw in a PUF-based authentication protocol, when outsider chosen perturbation security cannot be guaranteed. (ii) We present a solution to this flaw by introducing CBKA with an improved definition. (iii) We propose a provably secure and lightweight authentication protocol by combining PUFs and CBKA.

Keywords: Cryptographic protocol · Physically unclonable function · Channel-based key agreement

1 Introduction

The Internet of Things is on its way to change your everyday life. Computing devices are miniaturized and interconnected with an expected 50 billion devices by 2020 [13]. On the one hand, the IoT creates new multi billion markets for novel services and business models, but on the other hand, it also poses new challenges for security and privacy [4, 27]. Security-wise, the ubiquitous and dynamic nature of IoT drives the need for a strong identity management and, in particular, for secure device authentication. In addition to resource constraints, keys stored in non-volatile memory have to be assumed to be leaked, since attacks on these lightweight devices are hard to prevent [18]. With respect to privacy, sensor nodes and wearables, like fitness trackers, capture highly-personal data. Also, public-key cryptography may be too expensive to be implemented on resource-constrained

devices. In this paper, we answer how an IoT device can authenticate securely *and* privacy-friendly with said constraints.

For IoT platforms, physically unclonable functions [5, 14, 15, 31, 36] are an emerging trend which are often mentioned in the context of a lightweight solution. PUFs are already present on several products, such as small chip card microcontrollers like NXP’s SmartMX2 [1] to modern high-performance FPGAs [35]. When used as key storage, PUFs benefit from uncontrollable manufacturing variations causing an intrinsically embedded key to be unique. An often mentioned security advantage of PUF over, e.g. e-fuses, is that they store no values when the device is off [17].

PUFs are included in many authentication protocols. Delvaux *et al.* [9] surveyed multiple protocols using strong PUFs, and revealed that only few can offer privacy. Moriyama *et al.* [29] propose such a PUF-based authentication protocol under complete memory leakage. Their protocol was further adapted by Aysu *et al.* [6] by reversing the generate and reproduce procedures to suit better for lightweight devices at the cost of introducing a preshared secret. But helper data of reverse fuzzy extractors could leak information when a challenge is used multiple times [8], so we encrypt the helper data with a session key.

The lightweight and secure generation of a session key or shared secret can be addressed with CBKA [24–26], which uses the inherent randomness of a wireless communication channel between two devices, while offering information-theoretic security. CBKA ensures for each execution, that a fresh session key is generated due to its common physical randomness, so storage of preshared data becomes obsolete. A possible alternative to agree on a symmetric key is Diffie-Hellman over elliptic curves (ECDH), but we want to spare devices the rather heavy computational complexity and memory footprint of calculating on elliptic curves [38].

Both technologies – PUFs and CBKA – deal with physical noise and therefore need error correction and entropy amplification. Huth *et al.* [20] propose a system for the IoT integrating PUFs and CBKA alike. Their idea is to generate a symmetric key with CBKA and authenticate it with a PUF-based protocol. The implementation overhead is small since post-processing steps can be reused for either technology.

Contribution. We summarize our contribution as follows:

- *Flaw in existing protocol.* We show the need for fuzzy extractors with outsider chosen perturbation in Aysu’s protocol [6] to satisfy privacy claims, which are in conflict with lightweight and PUF-friendly fuzzy extractors.
- *Formal definition for CBKA.* We enhance their protocol, while keeping the reverse fuzzy extractor construction and possible usage of all fuzzy extractors. To prove security, we introduce a new formal definition for CBKA.
- *Protocol enhancement.* Our main contribution is a mutual authentication protocol enhancement. We integrate PUFs and CBKA, so that our protocol offers provable security and privacy under complete memory leakage assumptions and is suitable for the IoT. Summarizing, we shift digital challenges, i.e. key storage and session key generation, onto the physical domain.

Outline. First, mathematical preliminaries are introduced in Sect. 2 for notation, PUFs and fuzzy extractors. In Sect. 3, we describe our security and privacy model. Next, in Sect. 4 we show the flaw in an existing protocol. To overcome this flaw, we introduce a new formal definition for CBKA in Sect. 5. Based on our previous results, we present our main contribution in Sect. 6 – a provable secure and private mutual authentication protocol. In Sect. 7 we estimate implementation costs. We conclude this article in Sect. 8.

2 Notation and Preliminaries

We write \mathcal{M} to denote a metric space with an associated distance function dis . The statistical distance between two probability distributions A and B is denoted by $\text{SD}(A, B)$. U_n denotes the uniformly distributed random variable on $\{0, 1\}^n$. When A is a deterministic algorithm, $y := A(x)$ denotes the assignment to y from $A(x)$ with input x . When A is a probabilistic machine or algorithm, $y \xleftarrow{\text{R}} A(x)$ denotes that y is randomly selected from A according to its distribution. When A is a set, $y \xleftarrow{\text{U}} A(x)$ denotes that y is uniformly selected from A . $\tilde{H}_\infty(A)$ is the min-entropy of A and $\tilde{H}_\infty(A|B)$ indicates the conditional min-entropy of A given B . We denote an efficient algorithm as probabilistic polynomial time (PPT). We use a Truly Random Number Generator (TRNG) to derive truly random binary sequences. Furthermore, we use Symmetric Key Encryption $\text{SKE} := (\text{SKE.Enc}, \text{SKE.Dec})$, where SKE.Enc uses secret key sk and plaintext m as inputs and generates ciphertext c as output. SKE.Dec decrypts the ciphertext c with secret key sk to generate plaintext m . A Pseudorandom Function $\text{PRF} : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ inputs a secret key $sk \in \mathcal{K}$ and message $m \in \mathcal{D}$ providing an indistinguishable from random output for metric spaces $\mathcal{K}, \mathcal{D}, \mathcal{R}$.

Physically Unclonable Function. We adapt the definition by Armknecht *et al.* [3] for PUFs, which are parametrized by some thresholds δ_i , the number of iterations t , the number of inputs ℓ , the number of devices n , a negligible function $\epsilon(\cdot)$, and the security parameter λ . Here, we restate that a PUF is a probabilistic mapping $f : \mathcal{C} \rightarrow \mathcal{R}$ where \mathcal{C} is a domain space and \mathcal{R} is an output range of PUF f .

Requirement 1 (Intra-Distance [3]). *Whenever a single PUF is repeatedly evaluated with a fixed input, the maximum distance between the corresponding outputs is at most δ_1 . That is for any created PUF $f \leftarrow \mathcal{MP}(\text{param})$ and any $y \in \mathcal{C}$, it holds that $\Pr[\max(\{\text{dis}(z_i, z_j)\}_{i \neq j}) \leq \delta_1 | y \in \mathcal{C}, \{z_i \leftarrow f(y)\}_{1 \leq i \leq t}] = 1 - \epsilon(\lambda)$.*

Requirement 2 (Inter-Distance I [3]). *Whenever a single PUF is evaluated on different inputs, the minimum distance among them is at least δ_2 . That is for a created PUF $f \leftarrow \mathcal{MP}(\text{param})$ and for any $y_1, \dots, y_\ell \in \mathcal{C}$, we have $\Pr[\min(\{\text{dis}(z_i, z_j)\}_{i \neq j}) \geq \delta_2 | y_1, \dots, y_\ell \in \mathcal{C}, \{z_i \leftarrow f(y_i)\}_{1 \leq i \leq \ell}] = 1 - \epsilon(\lambda)$.*

Requirement 3 (Inter-Distance II [3]). Whenever multiple PUFs are evaluated on a single, fixed input, the minimum distance among them is at least δ_3 . That is for any created PUF $f_i \leftarrow \mathcal{MP}(\text{param})$ for $1 \leq i \leq n$ and any $y \in \mathcal{C}$, we have $\Pr[\min(\{\text{dis}(z_i, z_j)\}_{i \neq j}) \geq \delta_3 | y \in \mathcal{C}, \{z_i \leftarrow f_i(y)\}_{1 \leq i \leq n}] = 1 - \epsilon(\lambda)$.

Requirement 4 (Min-Entropy [3]). Whenever multiple PUFs are evaluated on multiple inputs, the min-entropy of the outputs is at least δ_4 , even if the other outputs are observed. Let $z_{i,j} \leftarrow f_i(y_j)$ be the output of a PUF f_i on input y_j where $f_i \leftarrow \mathcal{MP}(\text{param})$. Then $\Pr[\tilde{H}_\infty(z_{i,j} | \mathcal{Z}_{i,j}) \geq \delta_4 | y_1, \dots, y_\ell \in \mathcal{C}, \mathcal{Z} := \{z_{i,j} \leftarrow f_i(y_j)\}_{1 \leq i \leq n, 1 \leq j \leq \ell}, \mathcal{Z}_{i,j} := \mathcal{Z} \setminus \{z_{i,j}\}] = 1 - \epsilon(\lambda)$.

Definition 1 ([3]). A PUF $f : \mathcal{C} \rightarrow \mathcal{R}$ has $(\mathcal{MP}, t, n, \ell, \delta_1, \delta_2, \delta_3, \epsilon)$ -variance if the PUF's output has inter and intra distances as described in Requirements 1, 2 and 3, parameterized by $(\mathcal{MP}, t, n, \ell, \delta_1, \delta_2, \delta_3)$.

Definition 2 ([3]). A PUF $f : \mathcal{C} \rightarrow \mathcal{R}$ has $(\mathcal{MP}, n, \ell, \delta_4, \epsilon)$ -min-entropy if the PUF satisfies the min-entropy requirement as described in Requirement 4.

We define indistinguishability with a game as in [3]. The attacker is given access to all PUFs and can get any number of challenge-response pairs, except the two the attacker is choosing later. The information about all PUFs and challenge-response pairs is called state information st . The attacker cannot map a presented response back to one of two PUFs, even if the attacker chose these two PUFs and chose a challenge for each of these.

Definition 3 ([29]). Let \mathcal{A} be an adversary who can physically access PUFs f_i . Let \mathcal{S} be an algorithm which only interacts with f_i via oracle access. Let $z_{i,j} \leftarrow f_i(y_j)$ be the output of a PUF $f_i : \mathcal{C} \rightarrow \mathcal{R}$ on input y_j where $f_i \leftarrow \mathcal{MP}(\text{param})$. A PUF f_i satisfies $(\mathcal{MP}, n, \ell, \epsilon)$ -indistinguishability if for any distinguisher \mathcal{D} , the probability of distinguishing the outputs is negligibly close to ϵ such that $|\Pr[\mathcal{D}(1^\lambda, st) \rightarrow 1 | \{st \leftarrow \mathcal{A}(1^\lambda, f_i(y_j))\}_{1 \leq i \leq n, 1 \leq j \leq \ell}] - \Pr[\mathcal{D}(1^\lambda, st) \rightarrow 1 | \{st \leftarrow \mathcal{S}^{f_i(y_j)}(1^\lambda)\}_{1 \leq i \leq n, 1 \leq j \leq \ell}]] \leq \epsilon(\lambda)$.

Extractor. Strong extractors [30] allow to extract almost all min-entropy from a non-uniform random variable. Since we deal in this paper with secrets conditioned on some side information, we here recall the definition of average-case strong extractors, which are closely related to former strong extractors. Extractors guarantee that the extracted string is uniform, even when conditioned on a seed or any other external information.

Definition 4 (Average-case Extractor [11]). Let seed U_r be uniform on $\{0, 1\}^r$ and let X be any distribution over $\{0, 1\}^n$. Let E be any external information that may be correlated with X . A function $\text{Ext} : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ is an average-case (n, m, ℓ, ϵ) -strong extractor if for X with $\tilde{H}_\infty(X|E) \geq m$, we have $\text{SD}((\text{Ext}(X, U_r), U_r, E), (U_\ell, U_r, E)) \leq \epsilon$.

Definition 5 (Secure Sketch [11]). A (m, \tilde{m}, t) -secure sketch for \mathcal{M} is a pair of randomized procedures, “sketch” (SS) and “recover” (Rec), with the following properties:

- The sketching procedure SS on input $w \in \mathcal{M}$ outputs a bit string $s \in \{0, 1\}^*$. The recovery procedure Rec takes as inputs an element $w' \in \mathcal{M}$ and a bit string $s \in \{0, 1\}^*$.
- The correctness property of secure sketches guarantees that if $\text{dis}(w, w') \leq t$, then $\text{Rec}(w', \text{SS}(w)) = w$. If $\text{dis}(w, w') > t$, then no guarantee of the output of Rec can be given.
- The security property guarantees that for any distribution W over \mathcal{M} with $H_\infty(W) \geq m$, we have $\tilde{H}_\infty(W|\text{SS}(W)) \geq \tilde{m}$. The quantity \tilde{m} is the residual min-entropy and $m - \tilde{m}$ is the entropy loss of a secure sketch.

Definition 6 (Fuzzy Extractor [11]). A (m, ℓ, t, ϵ) -fuzzy extractor for \mathcal{M} is a pair of randomized procedures, “generate” (Gen) and “reproduce” (Rep), with the following properties:

- The generation procedure Gen on input $w \in \mathcal{M}$ outputs an extracted string $R \in \{0, 1\}^\ell$ and a helper string $P \in \{0, 1\}^*$. The reproduction procedure Rep takes $w' \in \mathcal{M}$ and bit string $P \in \{0, 1\}^*$ as inputs.
- The correctness property of fuzzy extractors guarantees that if $\text{dis}(w, w') \leq t$ and (R, P) is output by Gen(w), then $\text{Rep}(w', P) = R$. If $\text{dis}(w, w') > t$, then no guarantee of the output of Rep can be given.
- The security property guarantees that for any distribution W over \mathcal{M} of min-entropy m , with any external information E , the string R is close to uniform conditioned on P , i.e., if $H_\infty(W|E) \geq m$ and $(R, P) \leftarrow \text{Gen}(w)$, then $\text{SD}((R, P, E), (U_\ell, P, E)) \leq \epsilon$.

3 Security Model

Consider the two parties, a computationally powerful verifier \mathcal{V} and a resource-constrained prover \mathcal{P} , where \mathcal{P} is equipped with a PUF. The PUF is assumed to be premeasured for challenge-response pairs during the setup phase in a secure environment. In the key generation phase, verifier and prover agree on a secret session key via CBKA. In the authentication phase, they engage in a mutual authentication test. Upon acceptance, both parties output 1, or upon rejection, they output 0, thereby ending the session. Correctness requires to always accept session if all communications have been unaltered by adversary except with negligible probability of failure. Hence, we define security and privacy as in the work of Moriyama [29] and Aysu *et al.* [6].

Security. Intuitively, security requires that either of the legitimate nodes reject the session, when they detect a modified message by an adversary. In this paper, we assume a secure implementation of CBKA, a secure PUF and a secure fuzzy extractor on the prover, i.e., hardware and software Trojans, side-channel and fault

attacks and malware are outside the scope of this paper. Also, we assume the PUF to be tamper proof, i.e., tampering would modify its functionality, as it is standard for this security primitive. But, we do allow the adversary to issue a reveal query in the security game. This is a reasonable assumption as lightweight devices are prone to memory-leakage attacks during, e.g., the distribution chain. The adversary is also allowed to modify messages between verifier and prover at will. More formally, we consider the security game between a challenger and the adversary \mathcal{A} .

$$\begin{array}{l}
 \text{Exp}_{\Pi, \mathcal{A}}^{\text{Sec}}(k) \\
 \hline
 (pk, sk) \xleftarrow{R} \text{Setup}(1^k); \\
 sid^* \xleftarrow{R} \mathcal{A}_1^{\text{Launch, SendVerifier, SendProver, Result, Reveal}}(pk, \mathcal{R}, T); \\
 b := \text{Result}(sid^*); \\
 \text{Output } b
 \end{array}$$

In the defined security game, the adversary is able to issue following oracle queries $\mathcal{O} := (\text{Launch}, \text{SendVerifier}, \text{SendProver}, \text{Result}, \text{Reveal})$. The queries do:

- $\text{Launch}(1^k)$ Launch the verifier to initiate the session.
- $\text{SendVerifier}(m)$ Send arbitrary message m to the verifier.
- $\text{SendProver}(dev, m)$ Send arbitrary message m to the prover dev , where dev is the device with PUF $f_i(\cdot)$ and $1 \leq i \leq n$.
- $\text{Result}(sid)$ Output whether the verifier accepts the session sid , where sid is uniquely determined by the exchanged messages.
- $\text{Reveal}(dev)$ Output the secret key of the prover dev contained in the non-volatile memory.

The advantage of an active adversary \mathcal{A} against an authentication protocol Π is defined by probability $\text{Adv}_{\Pi, \mathcal{A}}^{\text{Sec}}(k)$ that $\Pr[\text{Exp}_{\Pi, \mathcal{A}}^{\text{Sec}}(k)]$ outputs 1 and the communication message in session sid^* is modified by the adversary \mathcal{A} . Note, the adversary can learn the memory content of the prover.

Definition 7 (Security, [29]). *An authentication protocol Π is secure against impersonation attack with complete memory leakage if for any PPT adversary \mathcal{A} , $\text{Adv}_{\Pi, \mathcal{A}}^{\text{Sec}}(k)$ is negligible in k (for large enough k).*

Privacy. In this paper, we use the privacy property from [29], which itself is based on the indistinguishability-based privacy model of the Juels-Weis privacy model [23]. Here, the adversary is allowed to issue the reveal query in any time to cover backward and forward privacy. However, there is a restriction that an honest protocol execution without adversarial influence is executed before and after the anonymous access. That way, prior and future tracing compromises can be locally neutralized and allow for some state update before and after the challenge is sent. The privacy model between the challenger and the adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ is recalled as follows.

$$\begin{array}{l}
\text{Exp}_{\Pi, \mathcal{A}}^{\text{IND}^*-b}(k) \\
\hline
(pk, sk) \xleftarrow{R} \text{Setup}(1^k); \\
(dev_0^*, dev_1^*, st_1) \xleftarrow{R} \mathcal{A}_1^{\mathcal{O}}(pk, \mathcal{R}, \mathcal{D}); \\
b \xleftarrow{U} \{0, 1\}, \mathcal{D}' := \mathcal{D} \setminus \{dev_0^*, dev_1^*\}; \\
\pi_0 \xleftarrow{R} \text{Execute}(\mathcal{R}, dev_0^*); \pi_1 \xleftarrow{R} \text{Execute}(\mathcal{R}, dev_1^*); \\
st_2 \xleftarrow{R} \mathcal{A}_2^{\mathcal{O}}(\mathcal{R}, \mathcal{D}', \mathcal{I}(dev_b^*), \pi_0, \pi_1, st_1); \\
\pi'_0 \xleftarrow{R} \text{Execute}(\mathcal{R}, dev_0^*); \pi'_1 \xleftarrow{R} \text{Execute}(\mathcal{R}, dev_1^*); \\
b' \xleftarrow{R} \mathcal{A}_3^{\mathcal{O}}(\mathcal{R}, \mathcal{D}, \pi'_0, \pi'_1, st_2); \\
b' := \text{Result}(sid^*); \\
\text{Output } b
\end{array}$$

As in the previous security game, we allow an adversary to interact with a verifier and a prover via oracle queries \mathcal{O} . Upon sending two devices (dev_0^*, dev_1^*) to the challenger, a random bit b is chosen and the adversary can access the challenge device dev_b^* anonymously. Upon issuing $\text{SendVerifier}(m)$, the challenger sends m to the challenge device dev_b^* and responds with its output. Also, as in the security game, the same holds true for the reveal query. We allow for re-synchronization opportunity before and after the anonymous access. The Execute query is the normal protocol execution between verifier and prover. The adversary can modify the transcript (π_0, π_1) , but not the communication, and (π'_0, π'_1) are given to the adversary. Concluding, the advantage of the adversary in guessing the correct device is defined as $\text{Adv}_{\Pi, \mathcal{A}}^{\text{IND}^*}(k) = |\Pr[\text{Exp}_{\Pi, \mathcal{A}}^{\text{IND}^*-0}(k) \rightarrow 1] - \Pr[\text{Exp}_{\Pi, \mathcal{A}}^{\text{IND}^*-1}(k) \rightarrow 1]|$.

Definition 8 (Privacy, [29]). *An authentication protocol Π satisfies the modified indistinguishability-based privacy under complete memory leakage if for any PPT adversary \mathcal{A} , $\text{Adv}_{\Pi, \mathcal{A}}^{\text{IND}^*}(k)$ is negligible in k (for large enough k).*

4 Flaw in Existing Protocol

Recall the protocol from Aysu *et al.* [6]. The weakness here is that an attacker can decrypt the helper data, as the symmetric key sk is stored in non-volatile memory, which can leak due to their defined security model. When an attacker also queries a device with the same challenge, e.g., by jamming the further protocol so that the key, as well as the challenge, gets never updated, he is able to get multiple helper data to this one challenge. Van Herrewege *et al.* [34] and Boyen [8] pointed out that this could leak information about the PUF. In this case, only fuzzy extractors providing outsider chosen perturbation security can be used, but these reusable fuzzy extractors are expensive to implement. Schaller *et al.* [32] address this issue with an extra post-processing step by adding a small amount of noise to the PUF response. However, we overcome this issue, If we do not rely on a preshared secret to encrypt the helper data. Rather we encrypt it with a freshly generated session key, as described in Sect. 6.

Verifier \mathcal{V}	Prover \mathcal{P} with $f_i(\cdot)$	
Setup phase		
$(sk, y_1) \stackrel{\text{U}}{\leftarrow} \text{TRNG}$	$\xrightarrow{sk, y_1}$ $\xleftarrow{z_1}$	$z_1 \stackrel{\text{R}}{\leftarrow} f_i(y_1)$
Authentication Phase		
<i>holds database</i> $\{(z_1, sk, z_{old}, sk_{old})\}_i$		<i>holds</i> $(f_i(\cdot), sk, y_1)$
$m_1 \stackrel{\text{U}}{\leftarrow} \text{TRNG}$	$\xrightarrow{m_1}$	$z'_1 \stackrel{\text{R}}{\leftarrow} f_i(y_1)$ $(r_1, hd) := \text{FE.Gen}(z'_1)$ $c := \text{SKE.Enc}(sk, hd)$
	$\xleftarrow{c, \dots}$	\vdots
$hd := \text{SKE.Dec}(sk, c)$		

Fig. 1. Protocol snippet by Aysu *et al.* If an adversary is able to get the secret key sk , the helper data hd can be decrypted from c by the same adversary.

The protocol snippet showing this flaw is illustrated in Fig. 1. Note, that the original protocol by Moriyama *et al.* [29] uses a forward fuzzy extractor and not a reverse one. Also in [29], the newly generated helper data gets encrypted with a fresh session key, which is not known before the authentication phase, due to previously exchanged nonces.

5 Channel-Based Key Agreement

Secure network communication relies on keys, preferably symmetric ones for efficiency reasons. ECDH is typically used in cryptographic protocols for the IoT. However, operations on an elliptic curve are computationally intense and require more energy compared to operations used for CBKA [38]. Additionally, CBKA agrees on a symmetric secret with information-theoretic security and therefore enables post-quantum security [25, 26]. This approach for key generation relies on the physical properties of the communication channel and essentially, it exploits three physical properties of multipath fading channels, namely reciprocity, temporal variation and spatial variation. These fundamental properties can be illustrated by the system model shown in Fig. 2 and are defined in Requirement 5.

Requirement 5 (Channel Observations). *Two legitimate nodes A and B generate a symmetric key from their respective channel observations $h_{BA}(t)$ and $h_{AB}(t)$ at time t . An adversary \mathcal{E} observes the communication between A and B so we have the following properties.*

- **Reciprocity:** *Considering the reciprocity property, it follows that the maximum distance between the channel observations of A and B at time t is at most δ_r , so it holds that $\Pr[\max(|h_{BA}(t) - h_{AB}(t)|) \leq \delta_r] = 1 - \epsilon$.*

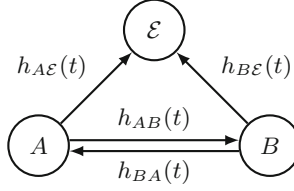


Fig. 2. Legitimate nodes A and B measure reciprocal properties of the physical channel at time t , denoted by $h_{BA}(t)$ and $h_{AB}(t)$. An adversary \mathcal{E} 's observations $h_{A\mathcal{E}}(t)$ and $h_{B\mathcal{E}}(t)$ are dependent on its relative position and are usually less correlated to $h_{BA}(t)$ and $h_{AB}(t)$ than $h_{BA}(t)$ is to $h_{AB}(t)$.

- **Coherence time:** Considering the coherence time τ_c , it follows that $\{h_{ij}(t) \approx h_{ij}(t + \delta_c)\}_{i,j \in \{A,B\}, i \neq j}$ with $\delta_c \leq \tau_c$.
- **Temporal variation:** Temporal variation introduces randomness, and thus, for properly chosen sampling time δ_s , it holds true that $\{h_{ij}(t) \neq h_{ij}(t + \delta_s)\}_{i,j \in \{A,B\}, i \neq j}$ with $\delta_s > \tau_c$.
- **Spatial variation:** Spatial variation prevents adversary \mathcal{E} from observing the same randomness as A and B for a sufficiently large distance with high probability, holding $\Pr \left[\{h_{ij}(t) \neq h_{i\mathcal{E}}(t)\}_{i,j \in \{A,B\}, i \neq j} \right] = 1 - \epsilon$.

Remark 1 (spacial decorrelation). The correlation between a node and an eavesdropper for a sufficiently large distance is often modelled on Jakes model [21]. Under this assumption the decorrelation is proportional to a zero-order Bessel function, where the first zero correlation occurs at approximately $\lambda/2$, with carrier wavelength λ [7]. For instance, $\lambda/2 \approx 6.25$ cm for the 2.4 GHz ISM frequency band. However, in practice a distance of $\lambda/2$ might not be enough as shown by Edmann *et al.* [12] and we note that secure distance determination is outside the scope of this paper.

CBKA is usually split up in different procedures, namely *channel measurement*, *quantization*, *information reconciliation* and *privacy amplification*. Measurement of the channel estimates the channel properties at a given time t , e.g., received signal strength indicator (RSSI) values can be used for this step, as these are available on almost all modern wireless transceivers. Next, the quantization takes the channel measurement $h_{ij}(t)$ at time t and assigns a digital value to this analog estimation. After quantization, the nodes A and B have similar bit strings with ideally no bit differences. There exist different approaches to sample the channel information and transform them into a bit string, suitable for further operations [2, 16, 22, 24, 33]. To allow for some errors during channel measurement and quantization due to physical noise, a reconciliation step is required, after which nodes A and B have the exact same bit string. During this information reconciliation both nodes exchange information, causing some entropy loss. Huth *et al.* [19] surveyed several information reconciliation schemes

for CBKA. As a last step, the privacy amplification [25], the entropy from the reconciled bit strings is compressed into a short key with nearly full entropy. For the notation of privacy amplification we adapt the definition of fuzzy extractors. Summarized, we define CBKA as follows:

Definition 9 (Channel-based Key Agreement). *Let A and B be two legitimate nodes, let \mathcal{E} be an adversary and let C_{CR} be an authentic common randomness that can be accessed by A , B and \mathcal{E} . A $(m_Q, m_{IR}, t, n, \ell, \epsilon)$ -channel-based key agreement is a tuple of procedures, CM, Q, IR and PA, with the following properties:*

- **Channel Measurement:** *The channel measurement procedure CM outputs channel observations $\{h_{ij}\}_{i,j \in \{A,B,\mathcal{E}\}, i \neq j}$ with accesses to an authentic common randomness C_{CR} . The channel observations h_{ij} need to fulfill requirement 5 and if $\{h_{ij} \leftarrow \text{CM}_j(C_{CR})\}_{i,j \in \{A,B,\mathcal{E}\}, i \neq j}$, then it holds for the mutual information that $I(h_{BA}, h_{A\mathcal{E}}) < I(h_{AB}, h_{BA}) > I(h_{AB}, h_{B\mathcal{E}})$.*
- **Quantization:** *The quantization procedure Q takes a channel observation $\{h_{ij}\}_{i,j \in \{A,B,\mathcal{E}\}, i \neq j}$ as input and outputs a string $\{q_j \in \{0,1\}^*\}_{j \in \{A,B,\mathcal{E}\}}$. If $\{q_j \leftarrow \text{Q}_j(\text{CM}_j(C_{CR}))\}_{j \in \{A,B,\mathcal{E}\}}$, then it holds that $H_\infty(q_A | (h_{A\mathcal{E}}, h_{B\mathcal{E}})) \geq m_Q \leq H_\infty(q_B | (h_{A\mathcal{E}}, h_{B\mathcal{E}}))$ and that $\text{dis}(q_A, q_{\mathcal{E}}) > \text{dis}(q_A, q_B) < \text{dis}(q_B, q_{\mathcal{E}})$.*
- **Information Reconciliation:** *The information reconciliation procedure IR takes as input a quantized string $\{q_i \in \mathcal{M}\}_{i \in \{A,B\}}$ from a legitimate node and outputs a reconciled strings $\{r_i \in \{0,1\}^n\}_{i \in \{A,B\}}$. The nodes A and B exchange helper strings s_A and $s_B \in \{0,1\}^*$. The correctness property of information reconciliation guarantees that if $\text{dis}(q_A, q_B) \leq t$, then $r_A = r_B$. The security property guarantees that for any distribution Q over \mathcal{M} with $H_\infty(Q) \geq m_Q$, we have $H_\infty(Q | (s_A \leftarrow \text{IR}_A(q_A, s_B), s_B \leftarrow \text{IR}_B(q_B, s_A), h_{A\mathcal{E}}, h_{B\mathcal{E}})) \geq m_{IR}$.*
- **Privacy Amplification:** *The privacy amplification procedure PA is an average-case $(n, m_{IR}, \ell, \epsilon)$ -strong extractor which takes an uniform seed I and a reconciled string $\{r_i \in \{0,1\}^n\}_{i \in \{A,B\}}$ from a legitimate node with min-entropy m_{IR} as inputs. The output key $\{k_i \in \{0,1\}^\ell\}_{i \in \{A,B\}}$ satisfies that $\{\text{SD}((\text{PA}_i(r_i, I), I), (U_\ell, I)) \leq \epsilon\}_{i \in \{A,B\}}$.*
- *The correctness property of channel-based key agreement guarantees that if the requirements for procedures CM, Q, IR and PA hold and the key is generated by $\{k_i \leftarrow \text{PA}_i(\text{IR}_i(\text{Q}_i(\text{CM}_i(C_{CR}))))\}_{i \in \{A,B\}}$, then $k_A = k_B$.*
- *The security property of channel-based key agreement guarantees that for any authentic common randomness C_{CR} , if the requirements for procedures CM, Q, IR and PA hold, the key k is close to uniform conditioned on all broadcasted information, i.e., $h_{A\mathcal{E}}, h_{B\mathcal{E}}, s_A$ and s_B .*

The quantity $m_Q - m_{IR}$ is defined as the *entropy loss* during information reconciliation IR. Note that parts of Definition 9 for CBKA are similar to Definitions 4, 5 and 6 of extractors, secure sketches and fuzzy extractors, respectively. This comes from the fact that information reconciliation of CBKA can be implemented with a secure sketch and privacy amplification can be implemented with

an extractor. Reusing these modules for PUFs and CBKA allows for a lightweight implementation on resource-constrained devices [20].

The correctness property of CBKA follows directly from the correctness property of information reconciliation. One instance of information reconciliation is a secure sketch with the code-offset construction for the Hamming metric as described in the work of Dodis *et al.* [11]. Here, reconciliation is successful if the used $(n, k, 2t+1)$ -code is able to correct the t Hamming errors occurring between the strings q_A and q_B . Clearly, this is always the case if t becomes not too big as required by $\text{dis}(q_A, q_B) \leq t$. For an in-depth analysis of secure sketches we refer the interested reader to the seminal paper by Dodis *et al.* [11].

Additionally, we assume CBKA with an authentic (unchanged) common randomness. To further strengthen our system against an attacker that can control the wireless channel, we can use a robust fuzzy extractor for information reconciliation as defined by Dodis *et al.* [10]. As an alternative, we can introduce an out-of-band (OOB) channel to guarantee authenticity of the channel during key agreement. Mirzadeh *et al.* [28] collected and compared multiple pairing protocols using OOB channels to exclude man-in-the-middle attacks.

6 Combined Protocol with PUF and CBKA

In this section we present our main contribution – an enhanced version of the protocol by Aysu *et al.* [6]. We attenuate the digital issue, as in Sect. 4, by shifting it to the physical world.

Our protocol is assumed to start in a secure setup phase, where a first challenge-response pair (y_1, z_1) is measured. The challenge y_1 is stored on the prover \mathcal{P} and the response is stored in the database hold by verifier \mathcal{V} .

After that, key generation phase via CBKA and authentication phase via PUF begins. A symmetric key sk is derived from the reciprocal channel between verifier \mathcal{V} and prover \mathcal{P} via CBKA as in Definition 9. The physical channel gets measured with CBKA.CM and is quantized with CBKA.Q. Information reconciliation CBKA.IR generates a secure sketch $s_{\mathcal{P}}$, with which the verifier \mathcal{V} can also derive the same string $r_{\mathcal{P}}$ as prover \mathcal{P} . Privacy amplification CBKA.PA can use some public randomness m_0 to extract the entropy in $r_{\mathcal{P}}$, resulting in the session key sk . We note again, that the common randomness C_{CR} is assumed to be authentic during the key generation phase or else no guarantee can be given about security. However, the security model allows the adversary to reveal the memory content before and after the key generation phase and authentication phase.

In the actual authentication phase, the helper data hd gets encrypted with the session key sk . The generation of pseudorandom values (s_1, \dots, s_4) follow from the FE.Gen output r_1 , and therefore from PUF response z_1 . To update for a new response, the device chooses a new challenge y_2 and one-time pads the response z_2 with s_2 . The value v_1 can be seen as a message authentication code, so that a manipulated c , m_2 or u_1 can be detected. Upon reception, the verifier decrypts the helper data hd and recovers his version of z_1 to the shared secret

r'_1 via FE.Rec. Next follows the generation of the same pseudorandom values as on the prover side and use s'_1 and s'_3 for verification purposes. If verification of v_1 holds then the database will get updated with the new response z'_2 . If not, the verification procedure repeats with the previously old response z_{old} . If this also fails s'_4 will be drawn randomly. On a successful database update, the verifier sends s'_4 to the prover, who updates his challenge $y_1 := y_2$ when s'_4 is valid. Our enhanced protocol is depicted in Fig. 3.

Theorem 1. *Let CBKA be a $(m_Q, m_{IR}, t, n, \ell, \epsilon)$ -channel-based key agreement as in Definition 9. Let $sk \leftarrow \text{CBKA}(C_{CR})$ be the output of CBKA for two legitimate nodes accessing an authentic common source of randomness C_{CR} . Let $f_i(\cdot)$ be a physically unclonable function, fulfilling Definitions 1, 2 and 3. Let $z_{i,j} \leftarrow f_i(y_j)$ be the output of a PUF $f_i : \mathcal{C} \rightarrow \mathcal{R}$ on input y_j where $f_i \leftarrow \mathcal{MP}(\text{param})$. Let FE be a (m, ℓ, t, ϵ) -fuzzy extractor as in Definition 6. Further assume that \mathcal{G} and \mathcal{G}' are secure pseudorandom functions. Then our protocol is secure against impersonation attacks with complete memory leakage as in Definition 7.*

Theorem 2. *Let CBKA be a $(m_Q, m_{IR}, t, n, \ell, \epsilon)$ -channel-based key agreement as in Definition 9. Let $sk \leftarrow \text{CBKA}(C_{CR})$ be the output of CBKA for two legitimate nodes accessing an authentic common source of randomness C_{CR} . Let $f_i(\cdot)$ be a physically unclonable function, fulfilling Definitions 1, 2 and 3. Let $z_{i,j} \leftarrow f_i(y_j)$ be the output of a PUF $f_i : \mathcal{C} \rightarrow \mathcal{R}$ on input y_j where $f_i \leftarrow \mathcal{MP}(\text{param})$. Let FE be a (m, ℓ, t, ϵ) -fuzzy extractor as in Definition 6. Further assume that \mathcal{G} and \mathcal{G}' are secure pseudorandom functions. Then our protocol holds the modified indistinguishability-based privacy under complete memory leakage as in Definition 8.*

The proof for Theorem 1 is given in Sect. A and the proof for Theorem 2 is stated in Sect. B.

7 Estimated Implementation Costs

As our proposed protocol in Fig. 3 is an enhancement of the original protocol in [6], there is only a marginal software and hardware overhead due to the integration of CBKA. Reference implementations of CBKA are on typical small microcontrollers as found in the IoT. However, the lightwightness of our proposed protocol comes at the cost of an expected execution time of roughly one minute.

Memory Footprint. Aysu *et al.* [6] provide two implementation results – one software implementation executed on a general purpose microcontroller and one with an additional hardware accelerator included. For the former one, they report a memory footprint of 8,104 bytes for text and 853 bytes for data on a MSP430, which offers a security of 128 bit. The data area includes global and local variables (stack, `bss` and `data`). For the latter one, they also list utilization of their

Verifier \mathcal{V}	Prover \mathcal{P} with $f_i(\cdot)$	
Setup phase		
$y_1 \xleftarrow{\text{U}} \text{TRNG}$	$\xrightarrow{y_1}$	$z_1 \xleftarrow{\text{R}} f_i(y_1)$
update database $(z_1, z_{old} := z_1)$	$\xleftarrow{z_1}$	
Key Generation Phase and Authentication Phase		
<i>Verifier \mathcal{V} and Prover \mathcal{P} have access to common randomness C_{CR}</i>		
holds database $\{(z_1, z_{old})\}_i$		holds $(f_i(\cdot), y_1)$
$h_{\mathcal{VP}} \xleftarrow{\text{R}} \text{CBKA.CM}(C_{CR})$		$h_{\mathcal{VP}} \xleftarrow{\text{R}} \text{CBKA.CM}(C_{CR})$
$q_{\mathcal{V}} := \text{CBKA.Q}(h_{\mathcal{VP}})$		$q_{\mathcal{P}} := \text{CBKA.Q}(h_{\mathcal{VP}})$
		$m_0 \xleftarrow{\text{U}} \text{TRNG}$
	$\xleftarrow{s_{\mathcal{P}}, m_0}$	$(r_{\mathcal{P}}, s_{\mathcal{P}}) := \text{CBKA.IR}(q_{\mathcal{P}})$
$r_{\mathcal{P}} := \text{CBKA.IR}(q_{\mathcal{V}}, s_{\mathcal{P}})$		
$sk := \text{CBKA.PA}(r_{\mathcal{P}}, m_0)$		$sk := \text{CBKA.PA}(r_{\mathcal{P}}, m_0)$
$m_1 \xleftarrow{\text{U}} \text{TRNG}$	$\xrightarrow{m_1}$	
		$z_1 \xleftarrow{\text{R}} f_i(y_1)$
		$(r_1, hd) := \text{FE.Gen}(z_1)$
		$c := \text{SKE.Enc}(sk, hd)$
		$m_2 \xleftarrow{\text{U}} \{0, 1\}^k$
		$(s_1, \dots, s_4) := \mathcal{G}(r_1, m_1 m_2)$
		$y_2 \xleftarrow{\text{U}} \{0, 1\}^k$
		$z_2 \xleftarrow{\text{R}} f_i(y_2)$
		$u_1 := s_2 \oplus z_2$
	$\xleftarrow{c, m_2, s_1, u_1, v_1}$	$v_1 := \mathcal{G}'(s_3, c m_2 u_1)$
$hd := \text{SKE.Dec}(sk, c)$		
$r'_1 := \text{FE.Rec}(z_1, hd)$		
$(s'_1, \dots, s'_4) := \mathcal{G}(r'_1, m_1 m_2)$		
check if $(s_1 = s'_1)$ for $1 \leq i \leq n$		
then verify $v_1 = \mathcal{G}'(s'_3, c m_2 u_1)$		
$z'_2 := s'_2 \oplus u_1$		
update database :		
$(z_1 := z'_2, z_{old} := z_1)$		
else $r'_1 := \text{FE.Rec}(z_{old}, hd)$		
$(s'_1, \dots, s'_4) := \mathcal{G}(r'_1, m_1 m_2)$		
\vdots		
else $s'_4 \xleftarrow{\text{U}} \{0, 1\}^k$	$\xrightarrow{s'_4}$	check if $(s_4 = s'_4)$
		update $y_1 := y_2$

Fig. 3. Detailed protocol of our proposed integration of PUF and CBKA.

hardware accelerator, which needs 3,543 lookup tables, 1,275 registers and 8 blocks of RAM on a Xilinx XC5VLX30-1FFG324. In summary, their implementation on a MSP430 with included hardware accelerator has a memory footprint of 4,920 bytes for text and 729 bytes for data on a MSP430, while also offering 128 bit of security. They note, that the hardware accelerator is about half the size as the MSP430 core. The data indicates that the protocol by Aysu *et al.* already fits into a small microcontroller.

With the implementation of Aysu *et al.* at hand, we only need to consider the additional CBKA implementation, i.e., implementation of the protocol steps CBKA.CM, CBKA.Q, CBKA.IR and CBKA.PA. The two steps of information reconciliation (CBKA.IR) and privacy amplification (CBKA.PA) form the construction of a fuzzy extractor, as described in Sect. 5. Therefore, these parts can be reused by the CBKA algorithm, resulting in only minor overhead for adapting the protocol state machine.

Channel measurements (CBKA.CM) are available if a wireless transceiver is present. However, if a wireless transceiver is present, measuring the channel results in no additional implementation cost, as every transceiver does so inherently. For example, Zenger *et al.* [37] implemented CBKA on an 8-bit Intel MCS-51, which is an SoC solution for the IoT. The authors state, that channel measurements are freely available on the given target platform. Also, their implementation offers a security of 128 bit.

The only algorithm that needs to be additionally implemented is the quantization (CBKA.Q). Zenger *et al.* report roughly 208 bytes resource overhead for their quantizer, which seems marginal compared to the original protocol implementation by Aysu *et al.*

Performance. Aysu *et al.* [6] state for their implementation of the original protocol, that it needs 111,965 to 1,730,922 clock cycles, depending on whether their proposed hardware accelerator is included. However, on their target constrained platform with a 1.846 Mhz clock, this results in an execution time on the device less than one second.

As described before, information reconciliation and privacy amplification can be reused from the implementation of Aysu *et al.*, which results in 18,597 to 690,174 additional clock cycles, depending if the hardware engine is included. This would result in a prolonged execution time of roughly half a second. Also, Zenger *et al.* report that their quantizer needs 11,876 clock cycles, which is negligible in terms of execution time.

However, regarding runtime the bottleneck is measuring the channel for CBKA, i.e. sampling enough entropy from the reciprocal channel. Here, Zenger *et al.* [37] state 60s for a 128-bit key agreement via CBKA.

8 Conclusion

With the proliferation and increased interconnection of lightweight IoT devices, security and privacy must not fall short. In this paper, we have shown security

is endangered when an existing authentication protocol is used with inexpensive, PUF-friendly fuzzy extractors. Our goal was to allow usage for these fuzzy extractors too, which offer no outsider chosen perturbation security. We achieved this with a new formal definition of CBKA and by enhancing an authentication protocol to fulfill all previously mentioned requirements. This paper shows how PUFs and CBKA can be securely integrated in the IoT, while avoiding costly public-key based solutions and associated public-key infrastructures.

A Security Proof

We use the proof provided by the work of Moriyama [29] and Aysu *et al.* [6] as a basis for our proof. The proof for Theorem 1 is as follows.

Proof. The adversary \mathcal{A} wants the verifier \mathcal{V} or the prover \mathcal{P} to accept the session while the communication is altered by the adversary. We concentrate only on the former case, as the verifier authentication is quite similar to that of the prover. We consider the following game transformations. Let S_i be the advantage that the adversary wins the game in Game i .

Game 0. This is the original game between the challenger and the adversary.

Game 1. The challenger randomly guesses the device dev^* with PUF $f_{i^*}(\cdot)$, where $i^* \xleftarrow{\mathcal{U}} \{1 \leq i \leq n\}$. If the adversary cannot impersonate dev^* to the verifier, the challenger aborts the game.

Game 2. Assume that ℓ is the upper bound of the sessions that the adversary can establish in the game. For $1 \leq j \leq \ell$, we evaluate or change the variables related to the session between the verifier and dev^* up to the ℓ -th session as the following.

Game 2- j -1. The challenger evaluates the output from the channel measurement and quantization of the CBKA algorithm implemented in dev^* at the j -th session. If the output does not have enough min-entropy m_Q or requirements for channel observations are violated, then the challenger aborts the game.

Game 2- j -2. The output from the information reconciliation procedure (r_P) is changed to a random variable.

Game 2- j -3. The output from the privacy amplification procedure (sk) is changed to a random variable.

Game 2- j -4. The challenger evaluates the output from the PUF implemented in dev^* at the j -th session. If the output does not have enough min-entropy m or requirements for intra-distance and inter-distance are violated, then the challenger aborts the game.

Game 2- j -5. The output from the fuzzy extractor (r_1) is changed to a random variable.

Game 2- j -6. The output from the PRF $\mathcal{G}(r_1, \cdot)$ is derived from a truly random function in this game.

Game 2- j -7. We change the PRF $\mathcal{G}(r_{old}, \cdot)$ to a truly random function.

Game 2-j-8. We change the XORed output $u_1 := s_2 \oplus z_2$ to randomly chosen $u_1 \xleftarrow{\mathcal{U}} \{0, 1\}^k$.

Game 2-j-9. The output from the PRF $\mathcal{G}'(s_3, \cdot)$ is derived from a truly random function in this game.

If the common source of randomness generates enough min-entropy, then the CBKA algorithm can output strings statistically close to uniform. Furthermore, if the PUF, that is equipped on the device generates enough min-entropy, then the fuzzy extractor can output strings statistically close to uniform. We then can set these strings as the seed for the PRF and the verifier and the prover share a common secret. So we can construct the challenge response authentication protocol with secure key update.

Lemma 1. $S_0 = n \cdot S_1$ (where n is the number of devices, i.e. provers).

Proof. If the adversary wins the game, there is at least one session which the verifier or prover accepts while the communication is modified by the adversary. Since the challenger randomly selects the session, the probability that the session is correctly guessed by the challenger is at least $1/n$.

Lemma 2. $|S_1 - S_{2-1-1}| \leq \epsilon$ and $|S_{2-(j-1)-9} - S_{2-j-1}| \leq \epsilon$ for any $2 \leq j \leq \ell$ if the CBKA algorithm is secure as required in Theorem 1.

Proof. Here, the output of the channel measurement and quantization of the CBKA algorithm has enough min-entropy and is independent from the other outputs except with negligible probability ϵ . If so, then there is no difference between these games. The property of CBKA assumed here says that even if the input to channel measurement and quantization of the CBKA algorithm is published, i.e. the authentic common randomness, the output derived from the input keeps the sufficient min-entropy property, and therefore each output is uncorrelated. Hence, the reveal query issued by the adversary is random looking by the assumption of this property.

Lemma 3. $|S_{2-j-1} - S_{2-j-2}| \leq \epsilon$ for any $2 \leq j \leq \ell$ if the CBKA.IR is an information reconciliation in a $(m_Q, m_{IR}, t, n, \ell, \epsilon)$ -channel-based key agreement.

Proof. Since we assumed that, always, the output from the quantization of the CBKA algorithm has enough min-entropy, the output of the information reconciliation procedure of the CBKA algorithm has enough min-entropy and is independent from the other outputs except with negligible probability ϵ . This is given by the security property of information reconciliation.

Lemma 4. $|S_{2-j-2} - S_{2-j-3}| \leq \epsilon$ for any $2 \leq j \leq \ell$ if the CBKA.PA is a privacy amplification in a $(m_Q, m_{IR}, t, n, \ell, \epsilon)$ -channel-based key agreement.

Proof. Since we assumed that, always, the output from the information reconciliation procedure of the CBKA algorithm has enough min-entropy, it is clear that no adversary can distinguish these games due to the randomization property of

privacy amplification, meaning privacy amplification guarantees that its output is statistically close to random. This is given by the security property of privacy amplification.

Lemma 5. $|S_{2-j-3} - S_{2-j-4}| \leq \epsilon \leq j \leq \ell$ if f is a secure PUF as required in Theorem 1.

Proof. Here, the PUF's output has enough min-entropy and is independent from the other outputs except with negligible probability ϵ . If so, then there is no difference between these games. The property of the PUF assumed here says that even if the input to the PUF is published, the output derived from the input keeps the sufficient min-entropy property, and therefore each output is uncorrelated. Hence, the reveal query issued by the adversary is random looking by the assumption of this property.

Lemma 6. $|S_{2-j-4} - S_{2-j-5}| \leq \epsilon$ for any $2 \leq j \leq \ell$ if the FE is a (m, ℓ, t, ϵ) -fuzzy extractor.

Proof. Since we assumed that, always, the output from the PUF has enough min-entropy, it is clear that no adversary can distinguish these games due to the randomization property of the fuzzy extractor, meaning the fuzzy extractor guarantees that its output is statistically close to random.

Lemma 7. $\forall 1 \leq j \leq \ell, |S_{2-j-5} - S_{2-j-6}| \leq \text{Adv}_{\mathcal{G}, \mathcal{B}}^{\text{PRF}}(k)$ where $\text{Adv}_{\mathcal{G}, \mathcal{B}}^{\text{PRF}}(k)$ is an advantage of \mathcal{B} to break the security of the PRF \mathcal{G} .

Proof. If there is a difference between these games, we construct an algorithm \mathcal{B} which breaks the security or PRF \mathcal{G} . \mathcal{B} can access the real PRF $\mathcal{G}(r_1, \cdot)$ or truly random function RF. \mathcal{B} sets up all secret keys and simulates our protocol except the n -th session. When the adversary invokes the n -th session, \mathcal{B} sends $m_1 \xleftarrow{\mathcal{U}} \{0, 1\}^k$ as the output of the verifier. When \mathcal{A} sends m_1^* to a device dev_i , \mathcal{B} selects m_2 and issues $m_1^* || m_2$ to the oracle instead of the normal computation of \mathcal{G} . Upon receiving (s_1, \dots, s_4) , \mathcal{B} continues the computation as the protocol specification and outputs (c, m_2, s_1, u_1, v_1) as the prover's response. When the adversary sends $(m_2^*, s_1^*, u_1^*, v_1^*)$, \mathcal{B} issues $m_1 || m_2^*$ to the oracle and obtains (s'_1, \dots, s'_6) .

If \mathcal{B} accesses the real PRF, this simulation is equivalent to Game 2-j-5. Otherwise, the oracle query issued by \mathcal{B} is completely random and this distribution is equivalent to Game 2-j-6. Thus we have $|S_{2-j-5} - S_{2-j-6}| \leq \text{Adv}_{\mathcal{G}, \mathcal{B}}^{\text{PRF}}(k)$.

Lemma 8. $\forall 1 \leq j \leq \ell, |S_{2-j-6} - S_{2-j-7}| \leq \text{Adv}_{\mathcal{G}, \mathcal{B}}^{\text{PRF}}(k)$.

Proof. The proof is as the proof for Lemma 7.

Lemma 9. $\forall 1 \leq j \leq \ell, S_{2-j-7} = S_{2-j-8}$.

Proof. Since the PRF $\mathcal{G}(r_1, \cdot)$ is already changed to the truly random function in Game 2-j-7, s_2 is used as effectively one-time pad to encrypt z'_2 . Therefore this transformation is purely conceptual change and the output distributions of these games are information theoretically equivalent.

Lemma 10. $\forall 1 \leq j \leq \ell, |S_{2-j-8} - S_{2-j-9}| \leq 2 \cdot \text{Adv}_{\mathcal{G}', \mathcal{B}'}^{\text{PRF}}(k).$

Proof. We can think that the seed input to the PRF \mathcal{G}' is changed to the random variable from the previous games. Consider an algorithm \mathcal{B} which interacts with PRF $\mathcal{G}'(s_3, \cdot)$ or random function RF. As in the proof for Lemma 7, \mathcal{B} simulates the protocol as the challenger up to the n -th session. \mathcal{B} generates (c, u_1) and issues $c||u_1$ to the oracle. \mathcal{B} generates the other variables as in the previous game and sends (c, m_2, s_1, u_1, v_1) as the prover's output after it obtains v_1 from the oracle. If the verifier receives $(c^*, m_2^*, s_1^*, u_1^*, v_1^*)$, \mathcal{B} checks that $(c^*, m_2^*, s_1^*) = (c, m_2, s_1)$. If so, \mathcal{B} issues $c^*||m_2^*||u_1^*$ to the oracle to check whether its response is identical to v_1^* .

If \mathcal{B} accesses the real PRF, this simulation is equivalent to Game 2- j -8. Otherwise, \mathcal{B} 's simulation is identical to Game 2- j -9. Thus the difference between these games are bounded by the security of PRF \mathcal{G}' .

Since the above game transformation is bounded by certain assumptions; i.e. for PUF, fuzzy extractor and PRFs, we can transform Game 0 to Game 2- ℓ -9. Considering Game 2- ℓ -9 there is no advantage for the adversary to impersonate the prover. Consider the case that the server accepts the session which is not actually derived the prover. Assume that the adversary obtains (c, m_2, s_1, u_1, v_1) from the prover. To mount the man-in-the-middle attack, the adversary must modify at least one of these variables.

Even when the adversary issues the reveal query and obtains y_1 before the session, he cannot predict the response z_1 . Since sk is generated after he can issue his reveal query, the session key remains secret and so hd remains encrypted. When the adversary modifies m_2 , the probability that the adversary wins the security game is negligible since s_1 is chosen from the truly random function. If m_2 is not changed, the verifier only accepts s_1 since it is deterministically defined by m_1 chosen by the verifier and m_2 . The first verification is passed only when the adversary reuses (c, m_2, s_1) , but v_1 is also derived from another random function. Thus the adversary cannot guess it and any modified message is rejected except with negligible probability. The same argument also applies to the verifier authentication, because the prover checks the verifier with the outputs from \mathcal{G} and \mathcal{G}' . Therefore, any adversary cannot mount the man-in-the-middle attack in our protocol and we finally have

$$\text{Adv}_{\Pi, \mathcal{A}}^{\text{Sec}}(1^k) \leq \frac{1}{2\ell n} \cdot \left(\text{Adv}_{\mathcal{G}, \mathcal{B}}^{\text{PRF}}(1^k) + \text{Adv}_{\mathcal{G}', \mathcal{B}'}^{\text{PRF}}(1^k) \right) + \epsilon$$

if the PUF and fuzzy extractor holds its properties.

B Privacy Proof

Again, we use the proof provided by the work of Moriyama [29] and Aysu *et al.* [6] as a basis for our proof. The proof for Theorem 2 is as follows.

Proof. The proof we provide here is similar to that for Theorem 1. However, we remark that it is important to assume that our protocol satisfies security as in Theorem 1 first for privacy to hold. The reason is that if the security is broken and a malicious adversary successfully impersonates device dev_0^* , the verifier will update the secret key that is not derived by the prover any more. So the verifier does not accept this prover after the attack and the adversary easily distinguishes the prover in the privacy game. Even if the adversary honestly transmits the communication message between $\mathcal{I}(dev_0^*)$ and the verifier in the challenge phase, the authentication result is always 0 and the adversary can realize which prover is selected as the challenge prover.

We modify Game 1 such that the challenger guesses two provers which will be chosen by the adversary in the privacy game. This probability that is at least $1/n^2$, and, then, we can continue the game transformation. After that, the game transformation described in Game 2 is applied to the sessions related to dev_0^* and dev_1^* . Then the communication message (c, m_2, s_1, u_1, v_1) and (s'_4) are changed to random variables. Even if the adversary can obtain the secret key of the prover within the privacy game, input to the PUF and helper data used in the challenge phase are independent from choices in the other phases. The re-synchronization allows this separation and new values are always random. Therefore, there is no information against which the adversary can distinguish the challenge prover in the privacy game, and we get:

$$\text{Adv}_{\Pi, \mathcal{A}}^{\text{IND}^*}(1^k) \leq \text{Adv}_{\Pi, \mathcal{A}'}^{\text{Sec}}(1^k) + \frac{1}{4\ell n^2} \cdot \left(\text{Adv}_{\mathcal{G}, \mathcal{B}}^{\text{PRF}}(1^k) + \text{Adv}_{\mathcal{G}', \mathcal{B}'}^{\text{PRF}}(1^k) \right) + \epsilon$$

for some algorithm $(\mathcal{A}', \mathcal{B}, \mathcal{B}')$ derived from the games.

References

1. NXP strengthens SmartMX2 security chips with PUF anti-cloning technology. <https://www.intrinsic-id.com/nxp-strengthens-smartmx2-security-chips-with-puf-anti-cloning-technology/>. Accessed 23 Aug 2016
2. Ambekar, A., Hassan, M., Schotten, H.D.: Improving channel reciprocity for effective key management systems. In: 2012 International Symposium on Signals, Systems, and Electronics (ISSSE), pp. 1–4. IEEE (2012)
3. Armknecht, F., Moriyama, D., Sadeghi, A.-R., Yung, M.: Towards a unified security model for physically unclonable functions. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 271–287. Springer, Heidelberg (2016). doi:[10.1007/978-3-319-29485-8_16](https://doi.org/10.1007/978-3-319-29485-8_16)
4. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
5. Aysu, A., Ghalaty, N.F., Franklin, Z., Yali, M.P., Schaumont, P.: Digital fingerprints for low-cost platforms using MEMS sensors. In: Proceedings of the Workshop on Embedded Systems Security, p. 2. ACM (2013)
6. Aysu, A., Gulcan, E., Moriyama, D., Schaumont, P., Yung, M.: End-to-end design of a PUF-based privacy preserving authentication protocol. In: Güneysu, T., Handschuh, H. (eds.) CHES 2015. LNCS, vol. 9293, pp. 556–576. Springer, Heidelberg (2015). doi:[10.1007/978-3-662-48324-4_28](https://doi.org/10.1007/978-3-662-48324-4_28)

7. Biglieri, E., Calderbank, R., Constantinides, A., Goldsmith, A., Arogyaswami Paulraj, H., Poor, V.: MIMO Wireless Communications. Cambridge University Press, New York (2007)
8. Boyen, X.: Reusable cryptographic fuzzy extractors. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 82–91. ACM (2004)
9. Delvaux, J., Peeters, R., Dawu, G., Verbauwhede, I.: A survey on lightweight entity authentication with strong PUFs. *ACM Comput. Surv.* **48**(2), 26: 1–26: 42 (2015)
10. Dodis, Y., Katz, J., Reyzin, L., Smith, A.: Robust fuzzy extractors and authenticated key agreement from close secrets. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 232–250. Springer, Heidelberg (2006). doi:[10.1007/11818175_14](https://doi.org/10.1007/11818175_14)
11. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004). doi:[10.1007/978-3-540-24676-3_31](https://doi.org/10.1007/978-3-540-24676-3_31)
12. Edman, M., Kiayias, A., Yener, B.: On passive inference attacks against physical-layer key extraction. In: Proceedings of the Fourth European Workshop on System Security, EUROSEC 2011, New York, NY, USA, pp. 8:1–8:6. ACM (2011)
13. Evans, D.: The internet of things: how the next evolution of the internet is changing everything. CISCO white paper, vol. 1, pp. 1–11 (2011)
14. Gassend, B., Clarke, D.E., van Dijk, M., Devadas, S.: Silicon physical random functions. In: Atluri, V. (ed.) Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, 18–22 November 2002, pp. 148–160. ACM (2002)
15. Guajardo, J., Kumar, S.S., Schrijen, G.-J., Tuyls, P.: FPGA intrinsic PUFs and their use for IP protection. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 63–80. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-74735-2_5](https://doi.org/10.1007/978-3-540-74735-2_5)
16. Guillaume, R., Ludwig, S., Müller, A., Czulwik, A.: Secret key generation from static channels with untrusted relays. In: 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 635–642 (2015)
17. Helfmeier, C., Nedospasov, D., Tarnovsky, C., Krissler, J.S., Boit, C., Seifert, J.-P.: Breaking and entering through the silicon. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, pp. 733–744. ACM (2013)
18. Herder, C., Meng-Day, Y., Koushanfar, F., Devadas, S.: Physical unclonable functions and applications: a tutorial. *Proc. IEEE* **102**(8), 1126–1141 (2014)
19. Huth, C., Guillaume, R., Strohm, T., Duplys, P., Samuel, I.A., Güneysu, T.: Information reconciliation schemes in physical-layer security: a survey. *Comput. Netw.* **109**, 84–104 (2016)
20. Huth, C., Zibuschka, J., Duplys, P., Güneysu, T.: Securing systems on the Internet of things via physical properties of devices and communications. In: Proceedings of 2015 IEEE International Systems Conference (SysCon 2015), pp. 8–13, April 2015
21. Jakes, W.C., Cox, D.C. (eds.): Microwave Mobile Communications. Wiley-IEEE Press, New York (1994)
22. Jana, S., Premnath, S.N., Clark, M., Kasera, S.K., Patwari, N., Krishnamurthy, S.V.: On the effectiveness of secret key extraction from wireless signal strength in real environments. In: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, pp. 321–332. ACM (2009)

23. Juels, A., Weis, S.A.: Defining strong privacy for RFID. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **13**(1), 7 (2009)
24. Mathur, S., Trappe, W., Mandayam, N., Ye, C., Reznik, A.: Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In: *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pp. 128–139. ACM (2008)
25. Maurer, U.: Secret key agreement by public discussion from common information. *IEEE Trans. Inf. Theor.* **39**(3), 733–742 (1993)
26. Maurer, U., Wolf, S.: Information-theoretic key agreement: from weak to strong secrecy for free. In: Preneel, B. (ed.) *EUROCRYPT 2000*. LNCS, vol. 1807, pp. 351–368. Springer, Heidelberg (2000). doi:[10.1007/3-540-45539-6_24](https://doi.org/10.1007/3-540-45539-6_24)
27. Medaglia, C.M., Serbanati, A.: An overview of privacy and security issues in the internet of things. In: Giusto, D., Iera, A., Morabito, G., Atzori, L. (eds.) *The Internet of Things*, pp. 389–395 (2010)
28. Mirzadeh, S., Cruickshank, H., Tafazolli, R.: Secure device pairing: a survey. *IEEE Commun. Surv. Tutorials* **16**(1), 17–40 (2014)
29. Moriyama, D., Matsuo, S., Yung, M.: PUF-based RFID authentication secure and private under memory leakage. *Cryptology ePrint Archive*, Report 2013/712 (2013). <http://eprint.iacr.org/2013/712>
30. Nishan, N., Zuckerman, D.: Randomness is linear in space. *J. Comput. Syst. Sci.* **52**(1), 43–52 (1996)
31. Pappu, S.R.: Physical one-way functions. Ph.D. thesis. Massachusetts Institute of Technology (2001)
32. Schaller, A., Skoric, B., Katzenbeisser, S.: Eliminating leakage in reverse fuzzy extractors. *IACR Cryptology ePrint Archive* 2014/741 (2014)
33. Tope, M.A., McEachen, J.C.: Unconditionally secure communications over fading channels. In: *Military Communications Conference, MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force*, vol. 1, pp. 54–58. IEEE (2001)
34. Van Herrewege, A., Katzenbeisser, S., Maes, R., Peeters, R., Sadeghi, A.-R., Verbauwhede, I., Wachsmann, C.: Reverse fuzzy extractors: enabling lightweight mutual authentication for PUF-enabled RFIDs. In: Keromytis, A.D. (ed.) *FC 2012*. LNCS, vol. 7397, pp. 374–389. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-32946-3_27](https://doi.org/10.1007/978-3-642-32946-3_27)
35. Wild, A., Güneysu, T.: Enabling SRAM-PUFs on xilinx FPGAs. In: *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–4. IEEE (2014)
36. Willers, O., Huth, C., Guajardo, J., Seidel, H.: MEMS-based gyroscopes as physical unclonable functions. *Cryptology ePrint Archive*, Report 2016/261 (2016). <http://eprint.iacr.org/2016/261>
37. Zenger, C.T., Pietersz, M., Zimmer, J., Posielek, J.-F., Lenze, T., Paar, C.: Authenticated key establishment for low-resource devices exploiting correlated random channels. *Comput. Netw.* **109**, 105–123 (2016)
38. Zenger, C.T., Zimmer, J., Pietersz, M., Posielek, J.-F., Paar, C.: Exploiting the physical environment for securing the internet of things. In: *Proceedings of the 2015 New Security Paradigms Workshop*, pp. 44–58. ACM (2015)

Information Security and Cryptology – ICISC 2016
19th International Conference, Seoul, South Korea,
November 30 – December 2, 2016, Revised Selected
Papers
Hong, S.; Park, J.H. (Eds.)
2017, XVI, 351 p. 31 illus., Softcover
ISBN: 978-3-319-53176-2