Secret-key capacity regions for multiple enrollments with an SRAM-PUF

Citation for published version (APA):

Kusters, C. J., & Willems, F. M. J. (2019). Secret-key capacity regions for multiple enrollments with an SRAM-PUF. *IEEE Transactions on Information Forensics and Security*, 14(9), 2276-2287. [8626480].
<https://doi.org/10.1109/TIFS.2019.2895552>

DOI:

[10.1109/TIFS.2019.2895552](https://doi.org/10.1109/TIFS.2019.2895552)

Document status and date:

Published: 01/09/2019

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Secret-Key Capacity Regions for Multiple Enrollments with an SRAM-PUF

Lieneke Kusters, *Student Member, IEEE*, and Frans M.J. Willems, *Fellow, IEEE*,

Abstract—We introduce the multiple enrollment scheme for SRAM-PUFs. During each enrollment the binary power-on values of the SRAM are observed, and a corresponding key and helper data are generated. Each key can later be reconstructed from an additional observation and the helper data. The helper data do not reveal information about the keys to an attacker. It is our goal to use the additional enrollments to consecutively increase the entropy of the generated key material.

We analyze two alternative settings. First, we present a regular setting, where each additional key is independent of all previous keys. Secondly, we introduce a key-replacement setting, where instead of an additional independent key, a new key (of increased length) is generated that replaces the old key. We characterize the capacity regions for both settings. We show that the total achievable secret-key rate is equal to the mutual information between all enrollment observations and a single (reconstruction) observation.

We derive our results based on a statistical model for SRAM-PUF that has been proposed in the literature. This model implies a *permutation symmetry* property of SRAM-PUF which plays a key role in our proofs.

I. INTRODUCTION

A PHYSICAL Unclonable Function (PUF) is a function that is embodied by a physical device and that maps challenges to responses [1], [2]. These responses are unique and hard to predict, and therefore they can be used to identify and authenticate individual devices. Since its initial definition, many different constructions have been developed that have PUF-like properties [3], [4]. In the current work we focus on SRAM-PUF, which is a popular PUF construction and also one of the simplest circuits in silicon that can be used as PUF [5], [6]. The SRAM-PUF response is a binary observation vector that corresponds to the power-on values of the SRAM cells.

Since PUF responses are noisy, so-called helper-data algorithms are used to extract uniform and reliable keys [7]. We study the problem of (re-)generating a uniform key from noisy PUF observations from information theoretic perspective, and we note that it is equivalent to the secret-key agreement scheme presented in [8] and [9]. Here, we refer to this as a single-enrollment scheme, see Figure 1. The single-enrollment scheme considers two phases. An enrollment phase, where an encoder generates a secret key s based on a first observation x_1 of the PUF. Then, in the reconstruction phase, a decoder reconstructs the key $\hat{s}$ based on a second (noisy) observation x_2 of the same PUF. During enrollment, besides the key,

Fig. 1. Single enrollment scheme.

also a helper data m is generated that will help the decoder to reconstruct the correct key. This helper data is publicly communicated to the decoder, and therefore it should not reveal any information about the key to an attacker.

Usually, enrollment is only performed once, while reconstruction can be repeated many times. Furthermore, it is known that the maximum achievable secret-key rate (secret-key capacity) for such a system is equal to the mutual information between the observation on the encoder side and the observation on the decoder side. Clearly, we can increase the mutual information, and thus the secret-key capacity, by considering multiple observations on the encoder side and/or the decoder side. This has inspired us to define a new scenario, that we call the multiple enrollment scheme.

In the multiple enrollment scheme, multiple keys are generated based on multiple observations of an SRAM-PUF, see Figure 3 and 4. We assume that each key and corresponding helper data are generated consecutively based on a next observation of the SRAM-PUF. Furthermore, the encoders require no information about previous observations or previously generated keys. At any time, reconstruction of the current keys can be performed by a decoder based on another SRAM-PUF observation and the previously generated helper data. As the helper data are sent to the decoders over a public channel, the helper data should not reveal any information about the keys to an attacker. We present two variants of the multiple enrollment scheme and derive the corresponding secret-key capacity regions.

Our results show that codes exist for which the key rate increases after each consecutive enrollment. Therefore, such codes improve upon the existing helper-data algorithms. Moreover, our multiple enrollment codes can be used when the same device should be enrolled in multiple systems. Finally, we use a statistical model of SRAM-PUF [10] to calculate the multiple enrollment key rates that could be achieved in practice.

A. Related Work

Helper-data algorithms were first introduced for generating keys from noisy biometrics measurements in [11] and

L. Kusters and F.M.J. Willems are with the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands (e-mail: {c.j.kusters, f.m.j.willems}@tue.nl). This work was funded by Eurostars-2 joint programme with co-funding from the EU Horizon 2020 programme under the E! 9629 PATRIOT project and the E! 11897 RESCUE project.

[12]. In [5] a similar algorithm was presented for deriving cryptographic keys from SRAM-PUF observation vectors. The achievable performance of helper-data algorithms under various conditions is analyzed in [13]. Finally, the achievable performance of secret-key generation schemes based on biometric sources is analyzed from information theoretic perspective in [14], [15].

The challenge of multiple enrollment was first studied for fuzzy extractors in [16]. Boyen showed that the regular fuzzy extractor scheme is not secure in general when the scheme is re-used on multiple observations of the same biometric source. In [17], [18] it was shown that when the density of the one-probabilities of the SRAM-PUF is symmetric, both the fuzzy commitment scheme and the syndrome method remain secure in the case of repeated enrollments. This result was further extended to the temperature dependent SRAM-PUF model for the unbiased case in [19]. In [20], de-biasing techniques were proposed that ensure security of the helper-data scheme after multiple enrollments for biased SRAM. Note that in all these works multiple enrollment was considered as repeating the same enrollment procedure multiple times. Therefore, these scenario's are different from the work presented in this paper, where in each enrollment a different encoding method may be used.

In [21], Lai *et al.* analyze a multiple use case scenario for the secret-key agreement schemes. They assume that the same biometric observation is reused in multiple systems, and derive the achievable privacy-security trade-off, both for a scenario where the systems are jointly designed, and for a scenario with an incremental design. The secret-keys in both subsystems are independent and uniform. Note that in [21] the total achievable rate does not increase with the number of enrollments, but stays constant instead.

Recent work that considers multiple observations at the encoder and/or the decoder side to improve the performance of the secret-key agreement system, can be found in [22]. Günlü and Kramer evaluate the performance by deriving the corresponding key-leakage-storage region as a function of the number of observations. However, in contrast to our work, they only generate a single key and helper data.

To our best knowledge, increasing the secret-key capacity using multiple enrollments is only investigated in [23] and [24]. In [23], we presented the 2-enrollment scheme and found the secret-key capacity using linear codes. In [24], we presented a forgetful setting and derived the corresponding secret-key capacity for any number of enrollments.

In the current work, we extend our previous results for the 2-enrollment scheme [23], to a t -enrollment scheme, with any number t of enrollments. Furthermore, we introduce the key-replacement setting and derive its capacity region. The key-replacement setting is closely related to the forgetful setting that was presented in [24]. However, in the forgetful setting the previous keys were considered forgotten and thus security of previously generated keys was not required. In the key-replacement setting the previously generated keys are replaced, but still remain secure from any attacker. This is a stronger requirement than before, and it results in a different proof. Interestingly, despite the stronger requirement, the capacity

region for the key-replacement setting is the same as for the forgetful setting [24].

B. Contributions and Outline

Our main contribution here, is presenting the capacity region for the multiple enrollment scheme for any number of enrollments. We present two alternative versions of the multiple enrollment scheme: the regular setting where each additional key is independent of the previous keys, and a key-replacement setting where each additional key can have increased rate but need not be independent of previously generated keys. For our analysis we introduce a simplified representation of the SRAM-PUF model.

In Section II, we give a short introduction to SRAM-PUFs, present a statistical model, and introduce notation and definitions that are used in the rest of the paper. As a reference and starting point, we present the single enrollment scenario in Section III and give the corresponding secret-key capacity. In the Sections IV-IX, we present two multiple enrollment scenario's, and derive the corresponding capacity regions. Finally, we discuss our results and implications for practical applications in Section X. We conclude with a summary of the presented work in Section XI.

II. NOTATION AND SRAM-PUF STATISTICAL MODEL

An SRAM memory can be used to temporarily store variables during computation. When the power of the memory is turned off, the stored values are lost. After the power is turned back on, a random binary value occurs in each memory cell that we call the power-on value of the SRAM cell.

Small uncontrollable variations in the production process result in variations in the physical properties and thus a slightly different behavior of each SRAM cell. For example, some cells have a very high probability that a one is observed after power-on, while other cells may have a very high probability that a zero is observed. As these variations between the cells cannot be controlled, this results in a random, unpredictable, distribution of zeros and ones across all memory cells. The power-on values of all cells of an SRAM together form a binary pattern that is unique for each SRAM. Therefore, this binary pattern is considered as a digital fingerprint and can be used for authentication [5], [6]. We use the power-on values of SRAM to generate secret keys.

In the following sections we first introduce the notation that is used in the paper. Then, we present a statistical model for the power-on values of an SRAM-PUF that is based on [25] [10], and derive a property of SRAM-PUF that we call the *permutation symmetry*.

A. Notation and Definitions

We use lowercase letters to denote constants and realizations of random variables. We observe t binary power-on values of n SRAM cells. The i^{th} observation of the j^{th} SRAM cell is represented by $x_{ij} \in \{0, 1\}$. When the cell-index j is not relevant, we refer instead to x_i as the i^{th} observation of any cell. We use a bold symbol $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})$,

Fig. 2. SRAM-PUF observations model.

to represent the i^{th} observation of all n SRAM cells. We call $\mathbf{x}_i$ the i^{th} observation vector of the SRAM, and it is always a binary vector of length n . After t observations of the SRAM-PUF, we have a sequence of observation vectors $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t)$. Calligraphic letters are used for finite sets and $|\mathcal{I}|$ denotes the cardinality of the set $\mathcal{I}$.

We use uppercase letters to denote random variables. Let $(X_1, \dots, X_t)$ be a tuple of t random variables and $\mathcal{I} \subseteq [1 : t]$. The subtuple of random variables with indices from $\mathcal{I}$ is denoted by $X(\mathcal{I}) = (X_i : i \in \mathcal{I})$. Similarly, for random vectors $(\mathbf{X}_1, \dots, \mathbf{X}_t)$, we have that $\mathbf{X}(\mathcal{I}) = (\mathbf{X}_i : i \in \mathcal{I})$.

In this paper (conditional) entropy and mutual information are defined for binary random variables as in [26], [27], and the unit is bit. Furthermore, we define jointly (weakly) typical sets $\mathcal{A}_\epsilon^{(n)}(X_1 X_2 \dots X_t)$ as in [26].

B. SRAM-PUF Statistical Model

We observe the power-on values of an array of n SRAM cells t times, and store the values in a sequence of binary observation vectors $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t)$, with $\mathbf{x}_i$ the i^{th} observation vector. We model each cell with a hidden state variable $\theta_j \in [0, 1]$ that defines the probability that a one is observed as the power-on value of that cell. That is, the probability that a one is observed at time i in cell j is $\Pr(X_{ij} = 1 | \Theta_j = \theta_j) = \theta_j$. In our model, the state variables of the cells are assumed to be independent and identically distributed over the cells with distribution $p_\Theta(\theta)$, and the values are unknown to an observer. We have visualized the SRAM-PUF observations model in Figure 2. Our statistical model for the observation vectors corresponds to the model that was derived for SRAM-PUF in [25] and [10]. It was shown [10], [25] that this model closely matches the statistics of experimental data. Note that we do not take the temperature into consideration. We assume that either the temperature of the observations is unknown, or all observations happen at the same temperature.

We are now interested in the probability distribution of the sequence of observation vectors $(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t)$. First, we derive the probability distribution for a single cell. The t observations of the j^{th} cell correspond to the j^{th} column of the matrix in Figure 2, and we use col_j as a short notation. Now the probability of observing the binary sequence col_j in cell j , is

$$\Pr((X_{1j}, X_{2j}, \dots, X_{tj}) = \text{col}_j)$$

$$= \int_0^1 \theta^{w_H(\text{col}_j)} (1 - \theta)^{t - w_H(\text{col}_j)} p_\Theta(\theta) d\theta,$$

with $w_H(\text{col}_j)$ the Hamming weight of the observed sequence (the number of observed ones). Note that the probability only depends on the number of ones and zeros that have been observed in col_j , not on the order of these observations. Therefore, any permutation of the observations of the j^{th} SRAM cell has the same probability, e.g.

$$\begin{aligned} \Pr((X_{1j}, X_{2j}, \dots, X_{tj}) = (x_{1j}, x_{2j}, \dots, x_{tj})) \\ = \Pr((X_{1j}, X_{2j}, \dots, X_{tj}) = (x_{2j}, x_{tj}, \dots, x_{1j})), \end{aligned}$$

and we call this property the *permutation symmetry* of SRAM-PUF.

Since the hidden state variables are all independent and identically distributed over the SRAM cells, the observations of the different cells (represented by the columns in Figure 2) are independent and identically distributed. Therefore the probability distribution of the observation vectors is

$$\begin{aligned} \Pr((\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t) = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t)) \\ = \prod_{j=1}^n \Pr((X_{1j}, X_{2j}, \dots, X_{tj}) = \text{col}_j) \\ = \prod_{j=1}^n \int_0^1 \theta^{w_H(\text{col}_j)} (1 - \theta)^{t - w_H(\text{col}_j)} p_\Theta(\theta) d\theta. \end{aligned}$$

As before, the probability distribution does not depend on the order of observations, but only on the number of observed ones and zeros. Therefore the *permutation symmetry* property holds not only for multiple observations of a single cell, but also for multiple observations of the group of cells, the observation vectors.

Now, we can derive the following properties for the joint entropy and the typical set, respectively,

$$\begin{aligned} H(\mathbf{X}(\mathcal{I})) &\stackrel{(a)}{=} nH(X(\mathcal{I})) \stackrel{(b)}{=} nH(X(\mathcal{I}')), \\ \mathcal{A}_\epsilon^{(n)}(X(\mathcal{I})) &\stackrel{(b)}{=} \mathcal{A}_\epsilon^{(n)}(X(\mathcal{I}')), \end{aligned}$$

for all $\mathcal{I}, \mathcal{I}' \subseteq [1 : t]$ with $|\mathcal{I}| = |\mathcal{I}'|$. Where (a) follows from independence of the SRAM cells and (b) from the *permutation symmetry*.

In the presented schemes, besides the t observation vectors that are used for enrollment by the encoders, an additional observation vector is required for the reconstruction of the key(s) by the decoders. For the sake of clarity of the derivations, we represent all encoder observation vectors by the symbol $\mathbf{x}_{t+1}$. We note that $(t + 1)$ could actually be replaced by any time index, due to the *permutation symmetry* of SRAM-PUF. Furthermore, the observation vectors that are observed by the different decoders could all have different time-indexes.

III. SINGLE ENROLLMENT

We start by discussing the single enrollment scheme in Figure 1. First, an encoder constructs a secret key s and helper data m after observing an observation vector $\mathbf{x}_1$. Then, a decoder observing another vector $\mathbf{x}_2$, which is a noisy version of $\mathbf{x}_1$, uses the helper data m to reconstruct an estimate $\hat{s}$ of

the original key. The helper data m should reveal sufficient information to the decoder such that reconstruction of the key is correct with high probability. We assume that the helper data m is public, and therefore the helper data should not reveal any information about the key s to an attacker. An attacker cannot observe any observation vector x_i , and tries to obtain information about the key s from the public helper data m .

An $(|S|, |\mathcal{M}|, n)$ code for the single enrollment scheme, consists of

- A set of key values $\mathcal{S} = \{1, 2, \dots, |\mathcal{S}|\}$ and a set of helper data values $\mathcal{M} = \{1, 2, \dots, |\mathcal{M}|\}$.
- An encoding function $E : \{0, 1\}^n \rightarrow \mathcal{S} \times \mathcal{M}$ that assigns a secret s and helper data m to each observation vector x_1 .
- A decoding function $D : \{0, 1\}^n \times \mathcal{M} \rightarrow \mathcal{S}$ that assigns an estimate $\hat{s} \in \mathcal{S}$ to each received pair (x_2, m) of an observation vector and helper data.

We are interested in the achievable secret-key rate for the single enrollment scheme for SRAM-PUFs, where the achievable secret-key rate is defined as follows.

Definition 1. A secret-key rate R is called *achievable* in the single enrollment setting, if for all $\delta > 0$ and for all n large enough, there exist encoders and decoders such that

$$\begin{aligned} \Pr(\hat{S} \neq S) &\leq \delta, \\ \frac{1}{n} H(S) + \delta &\geq \frac{1}{n} \log_2 |\mathcal{S}| \geq R - \delta, \\ \frac{1}{n} I(S; M) &\leq \delta. \end{aligned}$$

The secret-key capacity C_1 for the single enrollment setting is the maximum achievable secret-key rate.

Theorem 1. A secret-key rate R is achievable in the 1-enrollment setting if and only if

$$R \leq I(X_1; X_2).$$

The secret-key capacity is equal to $C_1 = I(X_1; X_2)$.

The single enrollment setting is the same as the secret-key generation scenario studied by Ahlswede and Csiszar [8] and Maurer [9]. A proof of achievability and the converse of Theorem 1 can be found e.g. in [28], [29]. There, a random coding argument was used to show that codes exist that can achieve the secret-key capacity C .

In the following, we introduce a new scenario where, where instead of only a single enrollment, multiple enrollments are performed by the encoder. We derive the secret-key capacity region for this setting using techniques that are inspired by [28], [29].

IV. MULTIPLE ENROLLMENTS

In the multiple enrollments setting, shown in Figure 3, t consecutive enrollments are performed. During the first enrollment, encoder E_1 observes an observation vector x_1 and generates a secret s_1 and corresponding helper data m_1 . The helper data provides sufficient information such that a decoder D_1 can reconstruct the secret from an additional observation x_{t+1} . During the second enrollment, encoder E_2 generates

Fig. 3. Multiple enrollments scheme, with t consecutive enrollments.

a second secret s_2 and corresponding helper data m_2 based on another observation vector x_2 . Now both helper data m_1 and m_2 provide sufficient information such that a decoder D_2 can reconstruct both secrets from an additional observation x_{t+1} . In general, during the i^{th} enrollment, an i^{th} secret s_i and corresponding helper data m_i are generated based on an i^{th} observation x_i . A decoder that observes x_{t+1} can decode all of the generated secrets $(s_1, s_2, \dots, s_i)$ as long as it has received all helper data up to this point, $(m_1, m_2, \dots, m_i)$.

The helper data $(m_1, m_2, \dots, m_t)$ are considered public, and therefore they should not reveal any information about the keys to an attacker, who cannot observe any of the observation vectors. All secret keys must be uniformly distributed and independent of each other.

An $(|S_1|, |\mathcal{M}_1|, |S_2|, |\mathcal{M}_2|, \dots, |S_t|, |\mathcal{M}_t|, n)$ -code for the t enrollments scheme, consists of

- t sets of keys and helper data, with the i^{th} sets given by $\mathcal{S}_i = \{1, 2, \dots, |\mathcal{S}_i|\}$ and $\mathcal{M}_i = \{1, 2, \dots, |\mathcal{M}_i|\}$, respectively.
- t encoding functions, where the i^{th} encoding function $E_i : \{0, 1\}^n \rightarrow \mathcal{S}_i \times \mathcal{M}_i$ assigns a secret s_i and helper data m_i to each observation vector x_i .
- t decoding functions, where the i^{th} decoding function $D_i : \{0, 1\}^n \times \mathcal{M}_1 \times \mathcal{M}_2 \times \dots \times \mathcal{M}_i \rightarrow \mathcal{S}_1 \times \mathcal{S}_2 \times \dots \times \mathcal{S}_i$ assigns the estimated keys $\hat{s}_1 \in \mathcal{S}_1$, $\hat{s}_2 \in \mathcal{S}_2$, ..., and $\hat{s}_i \in \mathcal{S}_i$ to each received tuple $(x_{t+1}, m_1, m_2, \dots, m_i)$ of any observation vector and i (previous) helper data..

We are interested in the achievable secret-key rate tuples for any number t of enrollments of an SRAM-PUF, where we assume that the rates are non-negative, and achievability is defined as follows.

Definition 2. A secret-key rate tuple $(R_1, R_2, \dots, R_t)$ is called *achievable* in the t -enrollment setting, if for all $\delta > 0$ and for all n large enough, there exist encoders and decoders such that

$$\begin{aligned} \Pr(\hat{S}_1 \neq S_1 \vee \hat{S}_2 \neq S_2 \vee \dots \vee \hat{S}_t \neq S_t) &\leq \delta, \\ \frac{1}{n} H(S_1 S_2 \dots S_t) + \delta &\geq \frac{1}{n} \log_2 |S_1| |S_2| \dots |S_t|, \\ \frac{1}{n} I(S_1 S_2 \dots S_t; M_1 M_2 \dots M_t) &\leq \delta, \end{aligned}$$

$$\frac{1}{n} \log_2 |\mathcal{S}_i| \geq R_i - \delta,$$

for all $i \in \{1, 2, \dots, t\}$.

The secret-key capacity region $\mathcal{C}^{(t)}$ of the t -enrollment setting is the set of all achievable rate tuples. The secret-key capacity is the maximum achievable rate for the i^{th} key, that is $C_i^t = \max_{(R_1, R_2, \dots, R_t) \in \mathcal{C}^{(t)}} R_i$. Finally, the sum-capacity is defined as the maximum achievable sum of the rates, $C_{\text{sum}}^t = \max_{(R_1, R_2, \dots, R_t) \in \mathcal{C}^{(t)}} R_1 + R_2 + \dots + R_t$.

The first inequality represents the condition of reliability, the second inequality represents the condition of uniformity and independence of the keys, the third inequality represents the leakage requirement. Finally, the last inequalities make a connection between the key-alphabet size and achievable rates. Note that the second inequality implies independence of the keys, since $\frac{1}{n}H(S_1) + \frac{1}{n}H(S_2) + \dots + \frac{1}{n}H(S_t) + \delta \geq \frac{1}{n}H(S_1 S_2 \dots S_t) + \delta \geq \frac{1}{n} \log_2 |\mathcal{S}_1| |\mathcal{S}_2| \dots |\mathcal{S}_t| \geq \frac{1}{n}H(S_1) + \frac{1}{n}H(S_2) + \dots + \frac{1}{n}H(S_t)$.

Theorem 2. The secret-key capacity region $\mathcal{C}^{(t)}$ of the t -enrollment setting is the set of all rate tuples $(R_1, R_2, \dots, R_t)$ such that

$$\begin{aligned} R_1 &\leq I(X_1; X_{t+1}), \\ R_1 + R_2 &\leq I(X_1 X_2; X_{t+1}), \\ &\vdots \\ R_1 + R_2 + \dots + R_t &\leq I(X_1 X_2 \dots X_t; X_{t+1}). \end{aligned}$$

The secret-key capacity $C_i^t = I(X_1 X_2 \dots X_i; X_{t+1})$, and the sum-capacity $C_{\text{sum}}^t = I(X_1 X_2 \dots X_t; X_{t+1})$.

In the next Sections, we first prove that all rate tuples that are inside the secret-key capacity region $\mathcal{C}^{(t)}$ are achievable. Furthermore, we prove the converse of Theorem 2, that is, no rate tuples are achievable that are outside of the capacity-region $\mathcal{C}^{(t)}$.

V. ACHIEVABILITY PROOF OF THEOREM 2

We prove that for any tuple $(R_1, R_2, \dots, R_t)$ inside the secret-key capacity region $\mathcal{C}^{(t)}$ of Theorem 2, codes exist such that all requirements in Definition 2 are satisfied. In our proof we make use of a random coding argument. Key in our proof is the virtual decoder, which can reconstruct all original observation vectors from all generated secrets and corresponding helper data.

We use $(2^{nR_{s_1}}, 2^{nR_{m_1}}, 2^{nR_{s_2}}, 2^{nR_{m_2}}, \dots, 2^{nR_{s_t}}, 2^{nR_{m_t}}, n)$ -codes for the t enrollments scheme as defined in Section IV. Here we have defined code rates $R_{s_i} = \frac{1}{n} \log_2 |\mathcal{S}_i|$ and $R_{m_i} = \frac{1}{n} \log_2 |\mathcal{M}_i|$ for $i \in \{1, 2, \dots, t\}$, that relate respectively the size of the key alphabet and the size of the helper-data alphabet to the length of the observation vectors.

The t sets of labelings that define the encoding functions $(E_1, E_2, \dots, E_t)$, are generated uniformly at random. Now, the first encoder maps an observation vector $\mathbf{x}_1$ to a secret key s_1 and helper data m_1 , according to the first encoding function E_1 . Similarly, the i^{th} encoder maps the observation vector $\mathbf{x}_i$ to a secret key s_i and helper data m_i , according to the i^{th} encoding function E_i .

We define t regular decoders and one virtual decoder. The regular decoders reconstruct secret keys from the received helper data and one observation of the SRAM values. The virtual decoder reconstructs all observation vectors $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t)$ that were observed during enrollment, using the helper data and secrets that were produced by the encoders. Note that the virtual decoder will not be implemented in practice. Instead, we use the fact that a virtual decoder can exist later in the proof, when we show that the secrets are uniform and that there is no leakage. In the following two subsections we describe the decoders in more detail.

A. Regular Decoders

The i^{th} decoder observes an observation vector $\mathbf{x}_{t+1}$ and all previous helper data $(m_1, m_2, \dots, m_i)$. Then he uses joint-typicality decoding to reconstruct the enrolled observation vectors $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i)$ and the corresponding secret $(s_1, s_2, \dots, s_i)$. This is explained in more detail in the following paragraphs.

The first decoder D_1 can reconstruct s_1 when only one helper data m_1 is available. The decoder finds the unique $\hat{\mathbf{x}}_1$ (and the corresponding key $\hat{s}_1$), such that $E_1(\hat{\mathbf{x}}_1) = (\hat{s}_1, m_1)$ and $(\hat{\mathbf{x}}_1, \mathbf{x}_{t+1}) \in \mathcal{A}_\epsilon^{(n)}(X_1 X_{t+1})$. Clearly, when decoding is successful and $\hat{\mathbf{x}}_1 = \mathbf{x}_1$, the secret is also correctly reconstructed. When no unique solution $\hat{\mathbf{x}}_1$ is found, the decoding fails and the decoder declares an error.

The second decoder D_2 has access to two helper data (m_1, m_2) and can reconstruct both corresponding keys (s_1, s_2) , using a two-step algorithm. First, it uses the same strategy as D_1 to reconstruct $\hat{\mathbf{x}}_1$ and $\hat{s}_1$. Then, it finds the unique $\hat{\mathbf{x}}_2$ (and the corresponding key $\hat{s}_2$), such that $E_2(\hat{\mathbf{x}}_2) = (\hat{s}_2, m_2)$ and $(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \mathbf{x}_{t+1}) \in \mathcal{A}_\epsilon^{(n)}(X_1 X_2 X_{t+1})$. Note that we use $\hat{\mathbf{x}}_2$ that has been reconstructed in the first step. When no unique solution is found, decoding fails and the decoder declares an error.

Similarly, each additional decoder D_i can reconstruct observation vector $\mathbf{x}_i$ and corresponding secret key s_i by sequentially decoding all previous observation vectors (and keys), starting by the first observation vector $\mathbf{x}_1$. For each subsequent decoding it uses the corresponding helper data, the observation vector $\mathbf{x}_{t+1}$, and all previously decoded observation vectors and applies joint typicality decoding. It decodes $\hat{\mathbf{x}}_i$ when $E_i(\hat{\mathbf{x}}_i) = (\hat{s}_i, m_i)$ and $(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_i, \mathbf{x}_{t+1}) \in \mathcal{A}_\epsilon^{(n)}(X_1 X_2 \dots X_i X_{t+1})$. When no unique solution is found, decoding fails and the decoder declares an error. When the decoder has successfully decoded the observation vector $\mathbf{x}_i$, the corresponding secret s_i follows automatically.

Instead of analyzing the probability of error, we notice that the problem of reconstructing $\mathbf{x}_i$ from $(m_1, m_2, \dots, m_i)$ and $\mathbf{x}_{t+1}$ corresponds to distributed lossless source coding with a helper. Therefore, it follows from Slepian-Wolf Theorem [30], [31] (see [27] Theorem 10.4) that the average error probability for the i^{th} decoder can be made negligibly small, when the helper-data rates satisfy

$$\sum_{j \in \mathcal{I}} R_{m_j} \geq H(X(\mathcal{I}) | X_{t+1} X(\mathcal{I}^c)) \text{ for all } \mathcal{I} \subseteq [1 : i].$$

We meet all requirements by setting $(R_{m_1}, R_{m_2}, \dots, R_{m_t})$ such that

$$R_{m_i} = H(X_i | X_{t+1} X_1 X_2 \dots X_{i-1}), \quad (1)$$

for all $i \in \{1, 2, \dots, t\}$. We define $\overline{P_{er}}$ as the probability, averaged over all possible uniformly generated encoding functions $(E_1, E_2, \dots, E_t)$, that any of the regular decoders fails. We conclude that $\overline{P_{er}} \rightarrow 0$, for rate tuples that satisfy (1) and for n large enough.

Note that (1) is also the optimal solution that meets the Slepian-Wolf requirements for all the regular decoders. This should become clear when we write all the requirements for the first two decoders. For the first decoder ($i = 1$), only one inequality needs to be satisfied:

$$R_{m_1} \geq H(X_1 | X_{t+1}).$$

It immediately follows that

$$R_{m_1} = H(X_1 | X_{t+1}),$$

is optimal. Then, for the second decoder ($i = 2$), three inequalities have to be satisfied:

$$R_{m_1} \geq H(X_1 | X_{t+1} X_2),$$

$$R_{m_2} \geq H(X_2 | X_{t+1} X_1),$$

$$R_{m_1} + R_{m_2} \geq H(X_1 X_2 | X_{t+1}).$$

The first requirement is already satisfied by our previous choice for R_{m_1} . Then, it follows that

$$R_{m_2} = H(X_2 | X_{t+1} X_1),$$

satisfies both the second and third requirement and therefore it is optimal. The same procedure can be repeated for the following decoders.

B. Virtual Decoder

In addition to the regular decoders we specify a virtual decoder. The virtual decoder reconstructs the observation vector tuple $(x_1, x_2, \dots, x_t)$ from all helper data $(m_1, m_2, \dots, m_t)$ and secret keys $(s_1, s_2, \dots, s_t)$. Note that virtual decoding is not actually taking place during key reconstruction. We will use existence of a virtual decoder in the final part of the proof, where we guarantee zero-leakage and uniformity of the secrets. The virtual decoder finds the unique tuple $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_t)$ such that $E_i(\tilde{x}_i) = (s_i, m_i)$ for all $i \in \{1, 2, \dots, t\}$ and furthermore $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_t) \in \mathcal{A}_e^{(n)}(X_1 X_2 \dots X_t)$. If such a unique tuple is not found an error is declared. We notice that the current coding problem corresponds to distributed lossless source coding, and it follows from Slepian-Wolf Theorem (see [27] Theorem 10.3) that the average error probability of the virtual decoder can be made negligibly small, when the rates satisfy,

$$\begin{aligned} \sum_{i \in \mathcal{I}} (R_{s_i} + R_{m_i}) &\geq H(X(\mathcal{I}) | X(\mathcal{I}^c)), \\ &\stackrel{(a)}{=} H(X_1 X_2 \dots X_{|\mathcal{I}|} | X_{|\mathcal{I}|+1} \dots X_t), \end{aligned}$$

for all $\mathcal{I} \subseteq [1 : t]$, and with $\mathcal{I}^c = [1 : t] \setminus \mathcal{I}$. Here (a) follows from *permutation symmetry* of the SRAM-PUF, see Section II.

Therefore, we select all rate tuples $(R_{s_1}, R_{s_2}, \dots, R_{s_t})$ such that

$$\sum_{i=1}^t R_{s_i} = I(X_1 X_2 \dots X_t; X_{t+1}), \quad (2)$$

$$\sum_{i=1}^k R_{s_i} \leq I(X_1 X_2 \dots X_k; X_{t+1}), \quad (3)$$

for all $k \in \{1, 2, \dots, t-1\}$.

In the following, we prove that all tuples that satisfy (1), (2), and (3), also satisfy the Slepian-Wolf requirements for the virtual decoder. First of all, for any subset $\mathcal{I} \subseteq [1 : t]$ of the rates, with maximum index k (that is $k = \max\{i : i \in \mathcal{I}\}$), we find

$$\begin{aligned} \sum_{i \in \mathcal{I}} (R_{s_i} + R_{m_i}) &\leq \sum_{i=1}^k R_{s_i} + \sum_{i \in \mathcal{I} \setminus k} R_{m_i} + R_{m_k} \\ &\stackrel{(a)}{\leq} I(X_1 X_2 \dots X_k; X_{t+1}) + \sum_{i=1}^{|\mathcal{I}|-1} R_{m_i} \\ &\quad + H(X_k | X_{t+1} X_1 X_2 \dots X_{k-1}) \\ &\stackrel{(b)}{=} H(X_{t+1}) + H(X_1 X_2 \dots X_{|\mathcal{I}|-1} | X_{t+1}) \\ &\stackrel{(c)}{=} H(X_1 X_2 \dots X_{|\mathcal{I}|}), \end{aligned} \quad (4)$$

where (a) follows from the fact that $R_{m_1} \geq R_{m_2} \dots \geq R_{m_t}$, and (b) and (c) follow from the *permutation symmetry* of the SRAM-PUF. Secondly,

$$\begin{aligned} \sum_{i=1}^t (R_{s_i} + R_{m_i}) &= I(X_1 X_2 \dots X_t; X_{t+1}) + H(X_1 X_2 \dots X_t | X_{t+1}) \\ &= H(X_1 X_2 \dots X_t). \end{aligned} \quad (5)$$

Finally, it follows from (4) and (5), that

$$\begin{aligned} \sum_{i \in \mathcal{I}} (R_{s_i} + R_{m_i}) &= \sum_{i=1}^t (R_{s_i} + R_{m_i}) - \sum_{i \in \mathcal{I}^c} (R_{s_i} + R_{m_i}) \\ &\geq H(X_1 X_2 \dots X_t) - H(X_1 X_2 \dots X_{|\mathcal{I}^c|}) \\ &= H(X_1 X_2 \dots X_{|\mathcal{I}|} | X_{|\mathcal{I}|+1} \dots X_t), \end{aligned}$$

where the last equality follows from the *permutation symmetry* of the SRAM-PUF. This concludes the proof that the selected rates meet the Slepian-Wolf requirements.

It follows that the average error probability of the virtual decoder can be made negligibly small for the selected rates. We define $\overline{P_{ev}}$ as the probability, averaged over all possible uniformly generated encoding functions $(E_1, E_2, \dots, E_t)$, that the virtual decoder fails. We conclude that $\overline{P_{ev}} \rightarrow 0$, for rate tuples that satisfy (1)-(3), and for n large enough.

C. Code Existence

Finally, let us define $\overline{P_e}$ as the probability, averaged over all possible uniformly generated encoding functions $(E_1, E_2, \dots, E_t)$, that any of the decoders, regular or virtual, fails. Then, by the union bound, $\overline{P_e} \leq \overline{P_{er}} + \overline{P_{ev}}$. We conclude that $\overline{P_e} \rightarrow 0$ for all rate tuples that satisfy (1)-(3) and for n large enough. Therefore, a code exists that achieves these rates, and that has a total error probability P_e that is negligibly small.

D. Zero-Leakage and Uniformity of the Keys

We have shown that a exists for each $(R_{s_1}, R_{m_1}, R_{s_2}, R_{m_2}, \dots, R_{s_t}, R_{m_t})$ tuple that satisfies (1)-(3), such that both the regular decoders and the virtual decoder are successful, and $P_e \rightarrow 0$. Now, we show that such codes also meet the requirements of zero-leakage and uniformity of the keys in Definition 2. First, we derive the following inequality,

$$\begin{aligned} nH(X_1 X_2 \dots X_t) &= H(\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_t) \\ &\stackrel{(a)}{=} H(\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_t S_1 M_1 S_2 M_2 \dots S_t M_t) \\ &\stackrel{(b)}{\leq} H(S_1 S_2 \dots S_t) + H(M_1 M_2 \dots M_t) + 1 + tnP_e \\ &\stackrel{(c)}{\leq} H(S_1 S_2 \dots S_t) + nH(X_1 X_2 \dots X_t | X_{t+1}) + 1 + tnP_e. \end{aligned}$$

In (a) we used the fact that the secrets and helper data directly follow from the observation vectors. In (b) we used Fano's inequality for the virtual decoder to derive an upper bound for $H(\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_t | S_1 M_1 S_2 M_2 \dots S_t M_t)$. And in (c) we used $H(M_1 M_2 \dots M_t) \leq R_{m_1} + R_{m_2} + \dots + R_{m_t} = H(X_1 X_2 \dots X_t | X_{t+1})$, which follows from the chosen helper data rates.

Now for any $\delta > 0$ we obtain that $\frac{1}{n}H(S_1 S_2 \dots S_t) + \delta \geq I(X_1 X_2 \dots X_t; X_{t+1})$ for n large enough. Furthermore, by the choice of the secret-key rates, we have $\frac{1}{n} \log_2 |S_1| |S_2| \dots |S_t| = R_{s_1} + R_{s_2} + \dots + R_{s_t} = I(X_1 X_2 \dots X_t; X_{t+1})$. Thus the uniformity condition of the secret keys in Definition 2 holds.

The resulting leakage about all secrets $(s_1, s_2, \dots, s_t)$ from helper data $(m_1, m_2, \dots, m_t)$ is

$$\begin{aligned} &I(S_1 S_2 \dots S_t; M_1 M_2 \dots M_t) \\ &= H(S_1 S_2 \dots S_t) + H(M_1 M_2 \dots M_t) \\ &\quad - H(S_1 M_1 S_2 M_2 \dots S_t M_t) \\ &\stackrel{(a)}{\leq} nH(X_1 X_2 \dots X_t) \\ &\quad - H(\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_t S_1 M_1 S_2 M_2 \dots S_t M_t) \\ &\quad + H(\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_t | S_1 M_1 S_2 M_2 \dots S_t M_t) \\ &\stackrel{(b)}{\leq} nH(X_1 X_2 \dots X_t) - H(\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_t) + 1 + tnP_e \\ &= 1 + tnP_e. \end{aligned}$$

In (a) we used $H(S_1 S_2 \dots S_t) + H(M_1 M_2 \dots M_t) \leq n(R_{s_1} + R_{m_1} + R_{s_2} + R_{m_2} + \dots + R_{s_t} + R_{m_t}) = n(I(X_1 X_2 \dots X_t; X_{t+1}) + H(X_1 X_2 \dots X_t | X_{t+1})) = nH(X_1 X_2 \dots X_t)$, which follows from the chosen secret and helper data rates. And in (b) we used Fano's inequality for the virtual decoder as before.

Thus for n large enough, $\frac{1}{n}I(S_1 S_2 \dots S_t; M_1 M_2 \dots M_t) \leq \delta$ for any $\delta > 0$, which satisfies the leakage requirement.

E. Conclusion

From (2) and (3), we conclude that the achievable secret-key rates are

$$\begin{aligned} R_1 &\leq I(X_1; X_{t+1}), \\ R_1 + R_2 &\leq I(X_1 X_2; X_{t+1}), \end{aligned}$$

$$\begin{aligned} &\vdots \\ &\vdots \\ R_1 + R_2 + \dots + R_t &\leq I(X_1 X_2 \dots X_t; X_{t+1}). \end{aligned}$$

It directly follows that the secret-key capacity $C_i^t = I(X_1 X_2 \dots X_i; X_{t+1})$, and the sum-capacity $C_{\text{sum}}^t = I(X_1 X_2 \dots X_t; X_{t+1})$.

VI. CONVERSE PROOF FOR THEOREM 2

We prove that no secret-key rate tuple $(R_1, R_2, \dots, R_t)$ is achievable that lies outside of the secret-key capacity region $\mathcal{C}^{(t)}$ of Theorem 2. Achievability is defined as in Definition 2.

We derive an upper bound for the joint entropy of the first i keys (for any $i \in 1, 2, \dots, t$),

$$\begin{aligned} &H(S_1 S_2 \dots S_i) = I(S_1 S_2 \dots S_i; \mathbf{X}_{t+1} M_1 M_2 \dots M_i) \\ &\quad + H(S_1 S_2 \dots S_i | \mathbf{X}_{t+1} M_1 M_2 \dots M_i) \\ &\stackrel{(a)}{\leq} I(S_1 S_2 \dots S_i; M_1 M_2 \dots M_i) \\ &\quad + I(S_1 S_2 \dots S_i; \mathbf{X}_{t+1} | M_1 M_2 \dots M_i) + 1 + in\delta \\ &\stackrel{(b)}{\leq} H(\mathbf{X}_{t+1}) - H(\mathbf{X}_{t+1} | S_1 M_1 \dots S_i M_i) + 1 + (i+1)n\delta \\ &\stackrel{(c)}{\leq} H(\mathbf{X}_{t+1}) - H(\mathbf{X}_{t+1} | \mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_i) + 1 + (i+1)n\delta \\ &\leq nI(X_1 X_2 \dots X_i; X_{t+1}) + 1 + (i+1)n\delta. \end{aligned}$$

For the first inequality (a), we use Fano's inequality for the decoders $D_1, D_2, \dots, D_i$ to derive an upper bound for $H(S_1 S_2 \dots S_i | \mathbf{X}_{t+1} M_1 M_2 \dots M_i)$. In (b) we use the leakage (last) requirement of Definition 2, and furthermore we use the fact that conditioning reduces entropy. In (c) we use again the fact that conditioning reduces entropy, and that the secrets and helper data directly follow from the observation vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. Therefore $H(\mathbf{X}_{t+1} | S_1 M_1 \dots S_i M_i) \geq H(\mathbf{X}_{t+1} | S_1 M_1 \dots S_i M_i \mathbf{X}_1 \dots \mathbf{X}_i) = H(\mathbf{X}_{t+1} | \mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_i)$.

The upper bounds for the entropy of the keys, together with the definition of the secret-key rates and the uniformity condition in Definition 2, result in the following inequalities

$$\begin{aligned} R_1 + R_2 + \dots + R_i - i\delta &\leq \frac{1}{n}H(S_1 S_2 \dots S_i) + \delta \\ &\leq I(X_1 X_2 \dots X_i; X_{t+1}) + \frac{1}{n} + (i+2)\delta, \end{aligned}$$

for all $i \in \{1, 2, \dots, t\}$. Now with $\delta \downarrow 0$ and $n \rightarrow \infty$, we obtain the upper bounds for the achievable rates as given in Theorem 2.

VII. KEY-REPLACEMENT

In the multiple enrollment setting, an additional key is generated during each consecutive enrollment. Until now, we have required that all the generated keys are independent. Now, we introduce a variation on the original scheme where each new key is not (necessarily) independent of the previous keys. We say that the new key replaces the previous keys and we call this new setting the multiple-enrollment setting with key-replacement. Since the new key replaces the old key, each decoder will only reconstruct the most recent key.

In the t -enrollment setting with key-replacement, t consecutive enrollments are performed, see Figure 4. During

Fig. 4. t enrollments scenario with key-replacement.

enrollment i a secret s_i and helper data m_i are generated, based on an observation x_i . A decoder that observes x_{t+1} can decode the most recent secret s_i as long as it has received all helper data up to this point, $(m_1, m_2, \dots, m_i)$. We assume that the helper data are public, and therefore they should not reveal any information about the keys to an attacker, who cannot observe the SRAM-PUF. Finally, all secret keys must be uniformly distributed.

An $(|\mathcal{S}_1|, |\mathcal{M}_1|, |\mathcal{S}_2|, |\mathcal{M}_2|, \dots, |\mathcal{S}_t|, |\mathcal{M}_t|, n)$ code for the t -enrollment setting with key-replacement, consists of

- t sets of keys and helper data, with the i^{th} sets given by $\mathcal{S}_i = \{1, 2, \dots, |\mathcal{S}_i|\}$ and $\mathcal{M}_i = \{1, 2, \dots, |\mathcal{M}_i|\}$, respectively.
- t encoding functions, where the i^{th} encoding function $E_i : \{0, 1\}^n \rightarrow \mathcal{S}_i \times \mathcal{M}_i$ assigns a secret s_i and helper data m_i to each observation vector x_i .
- t decoding functions, where the i^{th} decoding function $D_i : \{0, 1\}^n \times \mathcal{M}_1 \times \mathcal{M}_2 \times \dots \times \mathcal{M}_i \rightarrow \mathcal{S}_i$ assigns an estimate $\hat{s}_i \in \mathcal{S}_i$ to each received tuple $(x_{t+1}, m_1, m_2, \dots, m_i)$ of an observation vector and i (previous) helper data.

We are now interested in the achievable secret-key rate tuples for any number t of enrollments, where we assume that the rates are non-negative and achievability is defined as follows.

Definition 3. A secret-key rate tuple $(R_1, R_2, \dots, R_t)$ is called *achievable in the t -enrollment setting with key-replacement*, if for all $\delta > 0$ and for all n large enough, there exist encoders and decoders such that

$$\begin{aligned} \Pr(\hat{S}_1 \neq S_1 \vee \hat{S}_2 \neq S_2 \vee \dots \vee \hat{S}_t \neq S_t) &\leq \delta, \\ \frac{1}{n} H(S_i) + \delta &\geq \frac{1}{n} \log_2 |\mathcal{S}_i|, \\ \frac{1}{n} I(S_i; M_1 M_2 \dots M_t) &\leq \delta, \\ \frac{1}{n} \log_2 |\mathcal{S}_i| &\geq R_i - \delta, \end{aligned}$$

for all $i \in \{1, 2, \dots, t\}$. The secret-key capacity region $\mathcal{C}_{\text{rep}}^{(t)}$ of the t -enrollment setting with key-replacement is the set of all achievable rate tuples. The secret-key capacity is the maximum achievable rate for the i^{th} key, that is $\mathcal{C}_{\text{rep},i}^{(t)} = \max_{(R_1, R_2, \dots, R_t) \in \mathcal{C}_{\text{rep}}^{(t)}} R_i$.

Theorem 3. The secret-key capacity region $\mathcal{C}_{\text{rep}}^{(t)}$ of the t -enrollment setting with key-replacement is the set of all rate tuples $(R_1, R_2, \dots, R_t)$ such that

$$\begin{aligned} R_1 &\leq I(X_1; X_{t+1}), \\ R_2 &\leq I(X_1 X_2; X_{t+1}), \\ &\vdots \\ R_t &\leq I(X_1 X_2 \dots X_t; X_{t+1}). \end{aligned}$$

The secret-key capacity $\mathcal{C}_{\text{rep},i}^{(t)} = I(X_1 X_2 \dots X_i; X_{t+1})$.

In the next Sections, we first prove that all rate tuples that are inside the secret-key capacity region $\mathcal{C}_{\text{rep}}^{(t)}$ are achievable. Furthermore, we prove the converse of Theorem 3, that is, no rate tuples are achievable that are outside of the capacity-region $\mathcal{C}_{\text{rep}}^{(t)}$.

VIII. ACHIEVABILITY PROOF OF THEOREM 3

The achievability proof for the t -enrollment setting with key-replacement is similar to the proof of Theorem 2. Therefore, we focus on the differences here. In contrast to Section V, we now need to proof uniformity and zero-leakage for each secret key separately. Therefore, we change the definition of the virtual decoders and we require t virtual decoders instead of one. Furthermore, the zero-leakage proof is a bit more involved than before.

We use $(2^{nR_{s_1}}, 2^{nR_{m_1}}, 2^{nR_{s_2}}, 2^{nR_{m_2}}, \dots, 2^{nR_{s_t}}, 2^{nR_{m_t}}, n)$ -codes for the t -enrollment setting with key-replacement as defined in Section VII. As before, we define t regular decoders that reconstruct secret keys from the received helper data and one observation of the SRAM values. Furthermore, we define t virtual decoders to help us show that the secrets are uniform and that there is no leakage.

A. Regular Decoders

As in Section V, each regular decoder D_i can reconstruct observation vector x_i using joint typicality decoding with all previous helper data $(m_1, m_2, \dots, m_i)$ and an observation vector x_{t+1} . The corresponding secret s_i follows from the encoding function E_i . We set again

$$R_{m_i} = H(X_i | X_{t+1} X_1 X_2 \dots X_{i-1}), \quad (6)$$

for all $i \in \{1, 2, \dots, t\}$.

B. Virtual Decoders

The t virtual decoders operate slightly different from the single virtual decoder in Section V. The i^{th} virtual decoder reconstructs the observation vector tuple $(x_1, x_2, \dots, x_i)$ from all previous helper data $(m_1, m_2, \dots, m_i)$ and the current secret key s_i (instead of all previous keys).

First, he finds the unique $\tilde{x}_i$ such that $E_i(\tilde{x}_i) = (s_i, m_i)$ and $\tilde{x}_i \in \mathcal{A}_\epsilon^{(n)}(X_i)$. We can use a random binning argument (see Slepian-Wolf Theorem) to show that codes exist such that decoding of $\tilde{x}_i$ is successful with high probability when the labelings of the encoding functions have rates such that,

$$R_{s_i} + R_{m_i} \geq H(X_i).$$

Therefore, we choose

$$\begin{aligned} R_{s_i} &= I(X_{t+1}X_1X_2 \dots X_{i-1}; X_i) \\ &\stackrel{(a)}{=} I(X_1X_2 \dots X_i; X_{t+1}), \end{aligned} \quad (7)$$

for all $i \in \{1, 2, \dots, t\}$ and where (a) follows from the permutation symmetry.

Then, the virtual decoder reconstructs all previous observations $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{i-1})$ using the same strategy as the regular decoders. That is, he uses joint typicality decoding with all previous helper data and the observation vector $\mathbf{x}_i$. If the regular decoders are successful, then this decoding step is also successful.

C. Zero-Leakage and Uniformity of the Keys

We have shown that codes exist for each $(R_{s_1}, R_{m_1}, R_{s_2}, R_{m_2}, \dots, R_{s_t}, R_{m_t})$ tuple that satisfies both (6) and (7), such that both the regular decoders and the virtual decoders are successful. That is, for these codes the error probability P_e goes to zero for large n . Now, we show that these codes also meet the requirements of zero-leakage and uniformity of the keys in Definition 3.

We can use similar derivations as in Section V to show

$$\begin{aligned} nH(X_1X_2 \dots X_i) &\leq H(S_i) + nH(X_1X_2 \dots X_i|X_{t+1}) \\ &\quad + 1 + inP_e, \end{aligned}$$

for all $i \in \{1, 2, \dots, t\}$. Where we now used Fano's inequality for the i^{th} virtual decoder to give us an upper bound for $H(\mathbf{X}_1\mathbf{X}_2 \dots \mathbf{X}_i|M_1M_2 \dots M_iS_i)$.

Now for any $\delta > 0$ we obtain that $\frac{1}{n}H(S_i) + \delta \geq I(X_1X_2 \dots X_i; X_{t+1})$ for n large enough. Furthermore, by the choice of the secret-key rates, we have $\frac{1}{n}\log_2|S_i| = R_{s_i} = I(X_1X_2 \dots X_i; X_{t+1})$. Thus the uniformity conditions of the secret keys in Definition 3 hold.

The resulting leakage about a secret s_i from helper data $(m_1, m_2, \dots, m_t)$ is

$$\begin{aligned} &I(S_i; M_1M_2 \dots M_t) \\ &= H(S_i) + H(M_1M_2 \dots M_t) - H(M_1M_2 \dots M_iS_i) \\ &\quad - H(M_{i+1}M_{i+2} \dots M_t|M_1M_2 \dots M_iS_i) \\ &\stackrel{(a)}{\leq} nH(X_1X_2 \dots X_i) + nH(X_{i+1} \dots X_t|X_1 \dots X_iX_{t+1}) \\ &\quad - H(\mathbf{X}_1\mathbf{X}_2 \dots \mathbf{X}_iM_1M_2 \dots M_iS_i) \\ &\quad + H(\mathbf{X}_1\mathbf{X}_2 \dots \mathbf{X}_i|M_1M_2 \dots M_iS_i) \\ &\quad - H(M_{i+1}M_{i+2} \dots M_t|M_1M_2 \dots M_iS_i\mathbf{X}_{t+1}) \\ &\stackrel{(b)}{\leq} nH(X_1X_2 \dots X_i) + nH(X_{i+1} \dots X_t|X_1 \dots X_iX_{t+1}) \\ &\quad - H(\mathbf{X}_1\mathbf{X}_2 \dots \mathbf{X}_i) + 1 + inP_e \\ &\quad - H(\mathbf{X}_{i+1} \dots \mathbf{X}_tM_{i+1} \dots M_t|M_1M_2 \dots M_iS_i\mathbf{X}_{t+1}) \\ &\quad + H(\mathbf{X}_{i+1}\mathbf{X}_{i+2} \dots \mathbf{X}_t|M_1M_2 \dots M_tS_i\mathbf{X}_{t+1}) \\ &\stackrel{(c)}{\leq} nH(X_{i+1} \dots X_t|X_1 \dots X_iX_{t+1}) + 1 + inP_e \\ &\quad - H(\mathbf{X}_{i+1} \dots \mathbf{X}_t|\mathbf{X}_1 \dots \mathbf{X}_i\mathbf{X}_{t+1}) + 1 + (t-i)nP_e \\ &= 2 + tnP_e. \end{aligned}$$

In (a) we used $H(S_i) + H(M_1M_2 \dots M_t) \leq n(R_{m_1} + R_{m_2} + \dots + R_{m_t} + R_{s_i}) = n(I(X_1X_2 \dots X_i; X_{t+1}) +$

$H(X_1X_2 \dots X_t|X_{t+1})) = nH(X_1X_2 \dots X_i) + nH(X_{i+1} \dots X_t|X_1 \dots X_iX_{t+1})$, which follows from the chosen secret and helper data rates. In (b) we used Fano's inequality for the i^{th} virtual decoder. And in (c) we used Fano's inequality for the t^{th} regular decoder. Furthermore, we used in both (a) and (c) that conditioning reduces entropy.

Thus for n large enough, $\frac{1}{n}I(S_i; M_1M_2 \dots M_t) \leq \delta$ for any $\delta > 0$, which satisfies the leakage requirement.

D. Conclusion

From (7) we conclude that the achievable secret-key rates for the t -enrollments setting with key-replacement are

$$\begin{aligned} R_1 &\leq I(X_1; X_{t+1}), \\ R_2 &\leq I(X_1X_2; X_{t+1}), \\ &\vdots \\ R_t &\leq I(X_1X_2 \dots X_t; X_{t+1}). \end{aligned}$$

It directly follows that the secret-key capacity $C_{\text{rep},i}^t = I(X_1X_2 \dots X_i; X_{t+1})$.

IX. CONVERSE PROOF FOR THEOREM 3

We prove that no secret-key rate tuple $(R_1, R_2, \dots, R_t)$ is achievable that lies outside of the secret-key capacity region $\mathcal{C}_{\text{rep}}^{(t)}$ of Theorem 3. Achievability is defined as in Definition 3.

We can use similar steps as in Section VI to find the following upper bound for the entropy of the i^{th} key (for any $i \in 1, 2, \dots, t$),

$$\begin{aligned} H(S_i) &= I(S_i; \mathbf{X}_{t+1}M_1M_2 \dots M_i) \\ &\quad + H(S_i|\mathbf{X}_{t+1}M_1M_2 \dots M_i) \\ &\leq nI(X_1X_2 \dots X_i; X_{t+1}) + 1 + 2n\delta. \end{aligned}$$

The upper bounds for the entropy of the keys, together with the definition of the secret-key rates and the uniformity condition in Definition 3, result in the following inequalities

$$R_i - \delta \leq \frac{1}{n}H(S_i) + \delta \leq I(X_1X_2 \dots X_i; X_{t+1}) + \frac{1}{n} + 3\delta,$$

for all $i \in \{1, 2, \dots, t\}$. Now with $\delta \downarrow 0$ and $n \rightarrow \infty$, we obtain the upper bounds for the achievable rates as given in Theorem 3.

X. DISCUSSION OF RESULTS

In this paper we consecutively generate multiple keys based on multiple observations of the same SRAM-PUF. In the following we first analyze the secret-key capacity region that we found for the regular setting and then for the setting with key replacement. Finally, we discuss the properties of the schemes with respect to practical applications.

A. Secret-Key Capacity Region for Multiple Enrollment Setting

We have defined the multiple enrollment scheme in Section IV and we have derived the corresponding secret-key capacity region. The secret-key capacity region $\mathcal{C}^{(3)}$ for the multiple enrollment setting with $t = 3$ enrollments is visualized in Figure 5. Any triplet of secret-key rates that corresponds to a point on or below the red surface is inside the capacity region.

First of all, we are interested in the total achievable secret-key rate for this scheme. Since the keys are independent and uniformly distributed, we can consider a total key that is the concatenation of all the generated keys. This total key is uniformly distributed and has a secret-key rate equal to the sum of the rates, $R_{\text{total}}^t = \sum_{i=1}^t R_i$. Therefore, the maximum achievable total key rate is equal to the sum-capacity C_{sum}^t defined in Definition 2. It follows from Theorem 2, that the total capacity is $C_{\text{sum}}^t = I(X_1 X_2 \dots X_t; X_{t+1})$. We conclude that the total capacity in general increases with each additional enrollment. The magnitude of the increase depends on the statistics of the source. We have calculated the values of the sum-capacity for an empirical statistical model of SRAM-PUF and visualized the result in Figure. 6. Clearly, all rate tuples that have $\sum_{i=1}^t R_i = C_{\text{sum}}^t$ achieve the total capacity. For $t = 3$ enrollments, this corresponds to all secret-key rate triplets that lie on the red surface in Figure 5.

Secondly, we are interested in the maximum achievable rate for the i^{th} key, or the secret-key capacity C_i^t . It follows from Theorem 2, that $C_i^t = I(X_1 X_2 \dots X_i; X_{t+1})$. Furthermore, secret-key capacity for the i^{th} key can only be achieved if all preceding keys have rate zero, that is $R_j = 0, \forall j < i$. For example, for $t = 3$, secret-key capacity for the second key is achieved by all rate-triplets that lie on the edge of the capacity region marked by the squares (connecting the points $(0, I(X_1 X_2; X_4), I(X_3; X_4|X_1 X_2))$ and $(0, I(X_1 X_2; X_4), 0)$) in Figure 5.

B. Secret-Key Capacity Region for Multiple Enrollment Setting with Key-Replacement

We have defined the multiple enrollment scheme with key-replacement in Section VII and we have derived the corresponding secret-key capacity region. The secret-key capacity region $\mathcal{C}_{\text{rep}}^{(3)}$ for $t = 3$ enrollments is visualized in Figure 5. Any triplet of secret-key rates that corresponds to a point on or below the blue surface is inside the capacity region. We can immediately see that the secret-key capacity region for the multiple enrollment setting with key replacement includes the capacity region for the regular setting. Therefore, more rate tuples are achievable in this case. However, what can we say about the total achievable secret-key rate, and the secret-key capacity?

First of all, we are interested in the total achievable secret-key rate for this scheme. Since all previous keys are replaced by the most recent key, the total rate at any time is equal to the rate of the last key, so $R_{\text{rep, total}}^t = R_t$. Note that it does not make sense to concatenate the keys (as we did for the regular setting), since the keys are not independent here. Now, the

Fig. 5. Secret-key Capacity Regions for 3-enrollments.

maximum achievable total key rate is equal to the secret-key capacity of the last key, and it follows from Theorem 3, that the total capacity is $C_{\text{rep, total}}^t = I(X_1 X_2 \dots X_t; X_{t+1})$. Note that this is equal to the total capacity for the regular setting. Clearly, all rate tuples that have $R_t = C_{\text{rep, total}}^t$ achieve the total capacity. For $t = 3$ enrollments, this corresponds to all secret-key rate triplets that lie on the blue surface in Figure 5.

Second, we are interested in the maximum achievable rate for the i^{th} key, or the secret-key capacity $C_{\text{rep, } i}^t$. It follows from Theorem 3, that $C_{\text{rep, } i}^t = I(X_1 X_2 \dots X_i; X_{t+1})$. In contrast to the regular scheme, the i^{th} key can have the maximum rate, independent of the values of the rates of the preceding keys. For example, in Figure 5, all rate-triplets that lie on the face of the blue cube that is marked with triangles (with vertices $(I(X_1; X_4), I(X_1 X_2; X_4), 0)$, $(0, I(X_1 X_2; X_4), 0)$, $(0, I(X_1 X_2; X_4), I(X_1 X_2 X_3; X_4))$, and $(I(X_1; X_4), I(X_1 X_2; X_4), I(X_1 X_2 X_3; X_4))$) achieve secret-key capacity for the second key.

C. Multiple Enrollment Scheme for Practical Applications

The multiple enrollment scheme may be used in practice to consecutively increase the secret-key rate of a system when more observations of the SRAM-PUF become available over time. Depending on the application, the encoder may generate an additional (independent) key, or may replace a previous key by a newly generated key. We have shown that for both scenario's the achievable rate of uniform (independent) key-material is equal to the mutual information between the observations at the encoders and a single observation at the decoder, $I(X_1 X_2 \dots, X_t; X_{t+1})$.

In the following, we first calculate the corresponding values for a statistical model of SRAM-PUF from the literature [10]. Then, we discuss two properties of the schemes that may be of interest for practical applications. Finally, we comment on our results for scenario's in which the *permutation symmetry* assumption may be invalid, and we discuss possible application of the multiple enrollment scheme in authentication protocols.

1) *Secret-key capacity values:* We have seen that both versions of the multiple enrollment scheme can achieve the same total secret-key capacity. But what are typical values

Fig. 6. Secret-key capacity ($C_{\text{sum}}^t = C_{\text{rep},t}^t$) as a function of the number of enrollments t .

of the secret-key capacity? In [10] a statistical model for SRAM-PUF was presented and it was shown that the statistics match closely to the empirical data. We use this model with the parameters ($\lambda_1 = 0.1213$, $\lambda_2 = 0.0210$), as found in [10], to calculate the secret-key capacity after t enrollments, see the resulting plot in Figure 6. After a single enrollment we find $I(X_1; X_{t+1}) = 0.811$ bit and after 5-enrollments $I(X_1 X_2 \dots X_5; X_{t+1}) = 0.902$ bit (an increase of 11%) and after 25-enrollments $I(X_1 X_2 \dots X_{25}; X_{t+1}) = 0.927$ bit (an increase of 14%).

It is also visualized in the plot that the capacity approaches an upper bound. This can be easily explained using the statistical model for SRAM-PUF as presented in Section II. Each SRAM-cell has a hidden state variable θ that defines the one-probability of that cell. We can now find the following limit for the achievable secret-key rate

$$\begin{aligned} C_\infty &= I(X_1 X_2 \dots; X_{t+1}) = H(X_{t+1}) - H(X_{t+1} | X_1 X_2 \dots) \\ &\stackrel{(a)}{=} H(X_{t+1}) - H(X_{t+1} | X_1 X_2 \dots \Theta) \\ &\stackrel{(b)}{=} H(X_{t+1}) - H(X_{t+1} | \Theta) = I(X_{t+1}; \Theta). \end{aligned}$$

Here (a) follows since $E[X_i] = \Theta$ and, by the law of large numbers, $\lim_{t \rightarrow \infty} \Pr(|\frac{1}{t} \sum_{i=1}^t x_i - E[X_i]| > \epsilon) = 0$. And (b) follows by the Markov chain $X_1, X_2, \dots, X_t \leftrightarrow \Theta \leftrightarrow X_{t+1}$. For the parameters found in [10] this upper bound is calculated to be 0.933.

2) *Revealing a subset of the keys*: For the regular multiple enrollment scheme, we require the secret keys to be independent. As a result, all remaining keys remain secure even if a subset of the keys (together with the helper data) is revealed. Therefore, we can share the keys with different parties, and for example use each key to enroll in a different system. Any party that has access to a subset of the keys cannot obtain any information about the other keys, even when he has access to the corresponding helper-data.

The above property does not hold for the multiple enrollment setting with key-replacement, since the keys are not independent in this case.

3) *Autonomous encoders*: In both the regular and the key-replacement scenario the encoders do not require information about the previous enrollments to construct the current key and

corresponding helper data. Therefore, each encoder operates autonomously.

4) *Permutation symmetry assumption*: We note that in practice an SRAM-PUF may change its statistical behavior over time. This may for example be the result of temperature or voltage changes, or the result of aging effects. In such cases we need a more accurate model that takes the time-dependent effects into account. We can then use similar techniques to the current work in order to derive secret-key capacity regions. However, the results will be different (and time-dependent), since the *permutation symmetry* assumption cannot be used to simplify the equations. This also results in more complicated derivations, and makes it hard (if not impossible) to generalize to any number t of enrollments.

5) *Authentication protocols, attacks*: Our multiple enrollment scheme may be used to develop new authentication protocols for IoT (Internet of Things) devices. Lightweight authentication protocols based on PUFs are e.g. described in [32] (for RFID) and [33] (for IoT). It follows from our results that our keys are secure against any attack, as long as only the helper data is revealed to an attacker. Therefore, any protocol that would embed a multiple enrollment scheme should ensure that no information is revealed about the observation vectors or the keys.

XI. SUMMARY OF RESULTS

We have analyzed the achievable secret-key rates in multiple enrollment systems. During each consecutive enrollment an additional observation of the source is used to generate an additional key and corresponding helper data. Two scenarios are analyzed, a regular setting where each additional key is independent of previously generated keys, and the key-replacement setting where the previous key is replaced by a new key of increased rate. Our results show that codes exist for which consecutively increase the key rate after each enrollment. As future work we want to design codes that can achieve the derived secret-key rates.

ACKNOWLEDGMENT

We thank Tanya Ignatenko for collaboration during the PATRIOT project which led to formulation of the multiple enrollment problem. We thank the anonymous reviewers for helpful comments and suggestions.

REFERENCES

- [1] R. Pappu, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, sep 2002.
- [2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *9th ACM Conf. Comput. and Commun. Security - CCS*. ACM Press, 2002.
- [3] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Inform. Security and Cryptography*. Springer Berlin Heidelberg, 2010, pp. 3–37.
- [4] C. Böhm and M. Hofer, *Physical Unclonable Functions in Theory and Practice*. Springer New York, 2013.
- [5] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Cryptogr. Hardw. Embed. and Systems - CHES*. Springer Berlin Heidelberg, 2007, vol. 4727, pp. 63–80.

- [6] D. E. Holcomb, W. P. Bursleson, and K. Fu, "Power-Up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, 2009.
- [7] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for PUF-based key generation: Overview and analysis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 6, pp. 889–902, jun 2015.
- [8] R. Ahlswede and I. Csiszár, "Common randomness in information theory and cryptography. i. secret sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121–1132, jul 1993.
- [9] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 733–742, may 1993.
- [10] R. Maes, "An Accurate Probabilistic Reliability Model for Silicon PUFs," in *15th Int. Workshop Cryptogr. Hardw. and Embed. Systems - CHES*, 2013, pp. 73–89.
- [11] J.-P. Linnartz and P. Tuyls, "New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates," in *4th Int. Conf. Audio-, Video-based Biometric Person Authentication - AVBPA*, 2003, pp. 393–402.
- [12] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," in *Advances Cryptology - EUROCRYPT*, 2004, pp. 523–540.
- [13] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M.-D. M. Yu, "Efficient fuzzy extraction of puf-induced secrets: Theory and applications," *Cryptology ePrint Archive*, Report 2015/854, 2015, <https://eprint.iacr.org/2015/854>.
- [14] T. Ignatenko and F. Willems, "Biometric systems: Privacy and secrecy aspects," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 4, pp. 956–973, dec 2009.
- [15] L. Lai, S.-W. Ho, and H. V. Poor, "Privacy-security trade-offs in biometric security systems-part i: Single use case," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 122–139, Mar. 2011.
- [16] J. Boyen, "Reusable cryptographic fuzzy extractors," in *11th ACM Conf. Comput. and Commun. Security - CCS*, no. October. New York, New York, USA: ACM Press, 2004, p. 82.
- [17] L. Kusters, T. Ignatenko, and F. M. J. Willems, "Behavior of Temperature Dependent SRAM-PUFs, and Consequences for Secret-Key Capacity," in *7th Joint WIC/IEEE Symp. Inf. Theory Signal Process. Benelux - SITB*, 2017.
- [18] L. Kusters, T. Ignatenko, F. M. J. Willems, R. Maes, E. van der Sluis, and G. Selimis, "Security of helper data schemes for SRAM-PUF in multiple enrollment scenarios," in *IEEE Int. Symp. Inf. Theory - ISIT*, Jun. 2017, pp. 1803–1807.
- [19] L. Kusters, O. Günlü, and F. M. J. Willems, "Zero Secrecy Leakage for Multiple Enrollments of Physical Unclonable Functions," in *8th Joint WIC/IEEE Symp. Inf. Theory Signal Process. Benelux*, 2018, pp. 119–127.
- [20] R. Maes, V. van der Leest, E. van der Sluis, and F. M. J. Willems, "Secure key generation from biased PUFs: extended version," *J. Cryptographic Eng.*, vol. 6, no. 2, pp. 121–137, 2016.
- [21] L. Lai, S.-W. Ho, and H. V. Poor, "Privacy-security trade-offs in biometric security systems-part ii: Multiple use case," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 140–151, mar 2011.
- [22] O. Gunlu and G. Kramer, "Privacy, secrecy, and storage with multiple noisy measurements of identifiers," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2872–2883, nov 2018.
- [23] L. Kusters and F. M. J. Willems, "Capacity of a dual enrollment system with two keys based on an SRAM-PUF," in *Int. Zurich Seminar Commun. - IZS*, 2018.
- [24] —, "Capacity of a Multiple Enrollment System Based on an SRAM-PUF: Forgetful Setting," in *IEEE Int. Symp. Inf. Theory - ISIT*, 2018, pp. 2624–2628.
- [25] R. Maes, P. Tuyls, and I. Verbauwhede, "A soft decision helper data algorithm for SRAM PUFs," in *IEEE Int. Symp. Inf. Theory*, 2009, pp. 2101–2105.
- [26] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd ed. John Wiley & Sons, 2006.
- [27] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.
- [28] T. Ignatenko and F. M. J. Willems, "On the Security of the XOR-Method in Biometric Authentication Systems," in *27th Symp. Inf. Theory Benelux*, 2006, pp. 197–204.
- [29] T. Ignatenko, "Secret-key rates and privacy leakage in biometric systems," 2009.
- [30] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [31] T. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 226–228, mar 1975.
- [32] P. Gope, J. Lee, and T. Q. Quek, "Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.
- [33] P. Gope and B. Sikdar, "Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices," *IEEE Internet of Things Journal*.

Lieneke Kusters received the B.Sc. and M.Sc. degree in Electrical Engineering from Eindhoven University of Technology, The Netherlands, in 2011 and 2014, respectively. She is currently pursuing the Ph.D. degree at the Information and Communication Theory Laboratory, Eindhoven University of Technology. Her research interests include information theoretic privacy and security and physical unclonable functions.

Frans M.J. Willems was born in Stein, The Netherlands, in 1954. He received the M.Sc. degree in Electrical Engineering from the Technische Universiteit Eindhoven, Eindhoven, The Netherlands, and the Ph.D. degree from the Katholieke Universiteit Leuven, Leuven, Belgium, in 1979 and 1982 respectively.

From 1979 to 1982 he was a research assistant at the Katholieke Universiteit Leuven. In 1982 he joined the Electrical Engineering Department of the Technische Universiteit Eindhoven, where he is a full professor now. His research contributions are in the areas of multi-user information theory, noiseless source coding, data-embedding, and biometrics. From 1999 - 2016 he was an advisor for Philips Research Laboratories, Eindhoven, for subjects related to information theory.

Dr. Willems received the Marconi Young Scientist Award in 1982. From 1988 to 1990, he served as Associate Editor for Shannon Theory for the IEEE Transactions on Information Theory. From 2002 to 2006 he was Associate Editor for Information Theory for the European Transactions on Telecommunications. He is co-recipient of the 1996 IEEE Information Theory Society Paper Award for a paper in which the Context-Tree Weighting Algorithm was proposed. From 1998 to 2000 he was a member of the Board of Governors of the IEEE Information Theory Society. Since 2005 Dr. Willems is an IEEE Fellow. He was Counselor of the IEEE Student Branch Eindhoven from 2007 to 2014, and was Chairman of the IEEE Benelux Chapter on Information Theory from 2007 to 2017. More recently Dr. Willems received a 2011 Best Paper Award from the IEEE Signal Processing Society for the paper "Biometric Systems: Privacy and Security Aspects." In 2014 - 2015 Dr. Willems was a Distinguished Lecturer of the IEEE Information Theory Society. Dr. Willems has contributed more than two hundred journal and conference papers. He holds several patents.