symmetry

MDPI

Article

Privacy Amplification Strategies in Sequential Secret Key Distillation Protocols Based on Machine Learning

Jelica Radomirović [1,2,*], Milan Milosavljević [2,3], Branko Kovačević [1,2] and Miloš Jovanović [4]

[1] School of Electrical Engineering, Belgrade University, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia
[2] Vlatacom Institute of High Technology, Milutina Milankovica 5, 11070 Belgrade, Serbia
[3] Technical Faculty, Singidunum University, Danijelova 32, 11000 Belgrade, Serbia
[4] Faculty of Information Technologies, Belgrade Metropolitan University, Tadeuša Košćuška 63, 11000 Belgrade, Serbia
* Correspondence: jelica.radomirovic@vlatacom.com

Abstract: It is well known that Renyi's entropy of order 2 determines the maximum possible length of the distilled secret keys in sequential secret key distillation protocols so that no information is leaked to the eavesdropper. There have been no attempts to estimate this key quantity based on information available to the legitimate parties to this protocol in the literature. We propose a new machine learning system, which estimates the lower bound of conditional Renyi entropy with high accuracy, based on 13 characteristics locally measured on the side of legitimate participants. The system is based on a prediction intervals deep neural network, trained for a given source of common randomness. We experimentally evaluated this result for two different sources, namely 14 and 6-dimensional EEG signals, of 50 participants, with varying advantage distillation and information reconciliation strategies with and without additional lossless compression block. Across all proposed systems and analyzed sources on average, the best machine learning strategy, called the hybrid strategy, increases the quantity of generated keys 2.77 times compared to the classical strategy. By introducing the Huffman lossless coder before the PA block, the loss of potential source randomness was reduced from 68.48% to a negligible 0.75%, while the leakage rate per one bit remains in the order of magnitude 10^{-4}.

Keywords: key distillation; advantage distillation; information reconciliation; CASCADE; EEG; machine learning; Renyi entropy; Huffman coding; deep neural networks

Citation: Radomirović, J.; Milosavljević, M.; Kovačević, B.; Jovanović, M. Privacy Amplification Strategies in Sequential Secret Key Distillation Protocols Based on Machine Learning. *Symmetry* **2022**, *14*, 2028. https://doi.org/10.3390/sym14102028

Academic Editors: Jian-Qiang Wang and Christos Volos

Received: 29 August 2022
Accepted: 22 September 2022
Published: 27 September 2022

1. Introduction

The advent of quantum computing has resulted in a deeper theoretical re-examination of all existing cryptographic mechanisms and their practical implementations. The result of these reviews is a return to symmetric cryptographic systems and information-theoretical security measures. The central principle of this approach is simple to formulate: a cryptographic system provides absolute secrecy of messages if, and only if, the uncertainty (entropy) of its secret key is not less than the uncertainty of messages [1]. Systems designed in this way are known to be resistant to the unlimited computing resources of adversaries and thus to cryptanalysis based on quantum computers [2]. The price to be paid is the production and distribution of an enormous quantity of secret cryptographic keys, which must meet the highest criteria of true randomness. Nowadays, solutions for the problem of generating and distributing secret keys at the same time are mainly based on the fundamental results of Ahlsvede and Csiszar [3], Maurer [4], and Csiszar and Narayan [5]. The basic idea of this approach is the extraction of keys in a distributed communication scenario based on a random source whose correlated components are available to participants. Depending on the location of the source of uncertainty, there are two different approaches: (i) source is independent of communication channels (the source model), and (ii) source is the communication channel itself (the channel model) [4].

In this paper, we will deal only with the source model and corresponding sequential key distillation strategy by public discussions (SKD) since it has many advantages for professional applications, primarily in the domain of military and special operations [6]:

- Secret keys cannot be established through physical distribution during special operations;
- If there are no pre-generated and distributed secret keys, it makes them impossible to compromise before the start of critical operations;
- The principle of risk minimization dictates that compromising one subsystem does not compromise the entire system. Accordingly, the SKA system should be independent of cryptographic and telecommunication modules. This fact excludes the use of the SKA channel model, favoring the SKA source model.

The critical block in SKD is the privacy amplification (PA), which minimizes the amount of information available to an eavesdropper. If the PA is based on hash functions, the eavesdropper conditional Renyi entropy of order 2 (ECRE2) of sequence shared by legitimate parties determines the maximum length of the secret keys so that no information is leaked to the eavesdropper, see Corollary 4 [7]. However, two key things are difficult to quantify in practice:

1. How much of the total available pure randomness is allocated to secret keys?
2. Is there any leakage toward eavesdropper and what is the real security margin of the generated secret keys?

The first category of common practice is based on adopting a single global lower bound for ECRE2 for a given source. This constant is then used to determine the output dimension of a universal class of hash functions for all individual realizations of the protocol [4,7–9]. It is clear that in this way, the optimal strategy based on the local lower bound of ECRE2 is replaced by a far suboptimal one. We will call this strategy the global lower bound (GLB) strategy. The consequences of the GLB strategy are as follows:

- Restrictive PA block gives a significantly smaller quantity of secret keys;
- Unreliable quantitative estimate of leaked information to an eavesdropper;
- Privacy amplification and information reconciliation always appear together, which requires a cross-design between these two stages [10].

The second category of common practice is based on the (usually wrong) assumption that the input sequence in the PA block is uncorrelated with the corresponding eavesdropper sequence. In that case, relying on the well-known hash leftover lemma (HLL) [11], the PA block is designed to eliminate only that information that is leaked through the public discussion channel [12–14]. We will call this strategy the HLL strategy. The consequences of the HLL strategy are as follows:

- Absence of quantitative measures for the amount of leaked information;
- Unreliable estimation of security margin for generated secret keys;
- The final criterion for the validity of the SKD is reduced to passing some standard package of statistical tests (e.g., NIST suite, [15]). The choice of tests and their interpretation is a complex problem. Relying only on them takes us away from information-theoretical security.

In this paper, we propose a new PA design methodology based on a machine learning system that estimates ECRE2 and its lower bound. The system is based on a predicting intervals regression deep neural network (PIDNN), trained for a given source of common randomness. Inputs to the system are features locally measured on the side of legitimate participants. The basic properties of this system are as follows:

- A precise estimate of the lower bound for ECRE2;
- Quantification of the security margin of the generated keys;
- Quantification of the amount of leaked information;
- Quantification of the unused randomness of a given source;
- Quantification of the gain of the chosen strategy in relation to any other PA strategy.

In this way, we address the key issues of PA design, mentioned in points 1 and 2. Quantifications of the unused randomness of a given source and quantification of the gain of the chosen strategy in relation to any other PA strategy are introduced for the first time in this field. The quantification of the unused randomness of a given source can also be seen as a contribution to the general field of Information and Communications Technologies for Sustainable Development Goals [16] and encourages the development of services for privacy preserving [17].

The price to be paid for all the advantages of the proposed system is the formation of training sets and the training of PIDNN for a given source. Since this part of the work is performed only once in the offline mode, the complexity of the PA block in the working mode increases only for the computing resources necessary for ECRE2 prediction based on the already trained PIDNN.

The theoretical basis of the proposed method is formulated in Theorems 2 and 3, Section 3. In this way, we remain within the framework of information-theoretic security, which guarantees, in advance, the appropriate performance and security margins of the generated secret keys.

Experimental evaluation was conducted for two different sources, i.e., 6-dimensional and 14-dimensional EEG signals from 50 participants, varying the different advantage distillation (AD) and information reconciliation (IR) SKD strategies. The results show a significant increase in the quantity of generated secret keys without compromising their cryptographic quality. For a 14-dimensional and 6-dimensional EEG source, amplification is from 3.6 to 4.6 times and from 1.35 to 1.85 times, respectively, depending on the IR strategy and the lossless compression block before PA. On average, across all proposed systems and analyzed sources, the best machine learning strategy, called the hybrid strategy, increases the quantity of generated keys by 2.77 times compared to the classical strategy. By introducing the Huffman lossless coder before the PA block, the loss of potential source randomness was reduced from 68.48% to a negligible 0.75%, while the leakage rate per one bit remains in the order of magnitude 10^{-4}.

The rest of the paper is organized as follows. Section 2 provides the basics of the SKD strategy for the source model.

In Section 3, the classic PA block design strategy is presented. The importance of local decision making based on ECRE2 and the fundamental role of its lower bound was indicated. Theorem 2 can be considered a reformulation of Corollary 4 from [7] under conditions of knowing ECRE2. Theorem 3 gives the conditions under which the PA block provides maximum uncertainty about established secret keys on the Eve side if the lower bound for ECRE2 is known with a given probability.

In Section 4, the ML system for predicting the lower bound of ECRE2 is presented. It consists of an interval prediction deep neural network with 11 or 13 inputs and three outputs, which provide a prediction for ECRE2, as well as the lower and upper bounds of the range in which its true value lies with a predefined probability. Having a reliable lower bound enables a more efficient design of the PA block. Global indicators of gains are introduced, both in terms of the efficiency of using the randomness of a given source, as well as in terms of the quantity of generated keys in comparison with classical PA systems.

In Section 5, the synthesis of the proposed system in the case of two sources of EEG signals of different dimensions is presented in detail. The training set was obtained by recording EEG signals for 50 participants, resulting in an ensemble of $2 \times 50 \times (2 \times 50 - 1)/2 = 4950$ different triplets (Alice, Bob, Eve). After appropriate quantization and serialization, binary sequences of 36,000 bits in length were obtained for each participant. Experimental metrics include estimation of key generation rate (KR), the amount of information leaked to Eve, NIST's tests of the cryptographic quality of keys, as well as the gains compared to classical systems.

In the sixth concluding section, the complexity of system implementation and its practical usability are discussed, as well as some open issues concerning the efficient estimate of ECRE2.

2. Sequential Key Distillation Strategy

Figure 1 shows a source model for SKA within a scenario in which three parties, Alice, Bob, and Eve, observe realizations of a DMS. Each of them receives their own set of observations. Let X, Y, and Z, be Alice's, Bob's, and Eve's observations, respectively. Alice's and Bob's goals are to agree on a secret key K, based on their observations X and Y, so that Eve has no information about it. A public authenticated noiseless communication channel is fully available to all parties, including Eve.

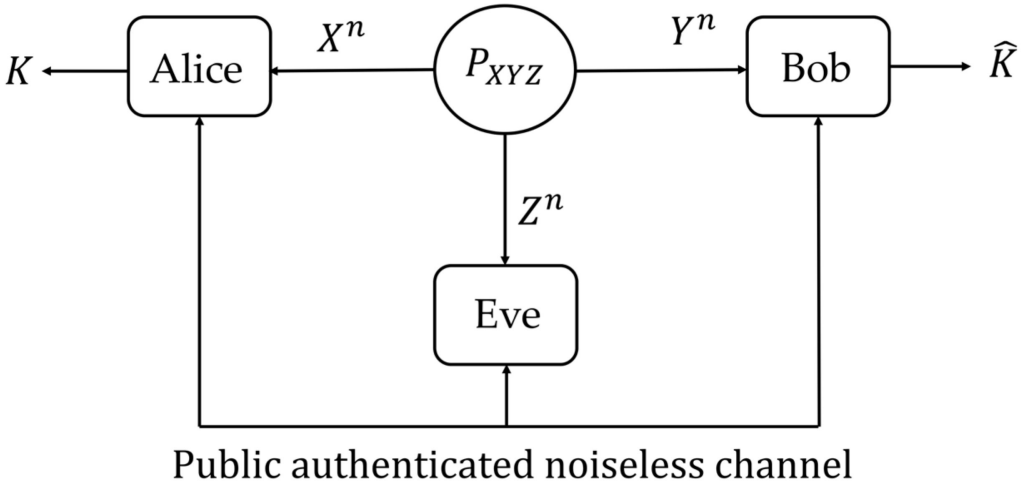

Figure 1. Source model for secret key agreement.

SKD strategy is four stage protocol consisting of [4]:

Randomness sharing. Alice, Bob, and Eve observe n realizations of DMS (XYZ, P_{XYZ}), where P_{XYZ} denotes the joint probability measure of the random variables X, Y, and Z.

Advantage distillation. If necessary, Alice and Bob exchange messages over a public channel to "distill" the observation parts on which they have an advantage over Eve.

Information reconciliation. Alice and Bob exchange messages over the public channel in order to eliminate mutual differences and agree on a common binary string.

Privacy amplification. Alice and Bob publicly agree on a deterministic function that they would apply to their common sequence to generate the final secret key.

The secrecy capacity of a public channel is the maximum rate at which information can be reliably exchanged between legitimate parties such that the rate at which an eavesdropper obtains this information is arbitrarily small.

The secret key capacity is, at the same time, the maximum length of a secret key that can be sent in the presence of an eavesdropper and can be defined by

$$C_k = min\{I(X;Y), I(X;Y|Z)\}, \quad (1)$$

where $I(X;Y)$ denotes the mutual information between X and Y, while $I(X;Y|Z)$ denotes this mutual information conditioned by Z. The advantage of the SKD strategy is the proven achievement of all secret key rates lower than the secrecy capacity C_k, as well as its explicit practical implementation [4].

3. PA Strategy

PA is the last stage of SKD in which all the information that was leaked to Eve in the PA and IR protocol execution phase is eliminated from the common sequence of Alice and Bob. As a rule, it is achieved by applying a suitable hash function from the class of universal hash functions or by applying a class of functions called extractors [18]. In the theoretical sense, both procedures are equivalent. Recent studies show that the use of extractors is superior to the use of hash functions only with the very large key length of order greater

than 10^5 bits [19]. Keeping in mind the typical practice, we will limit ourselves to PA based on hash functions from the universal class of hash functions, defined as follows [20]:

Definition 1. *Given two finite sets $\mathcal{A}$ and $\mathcal{B}$, a family $\mathcal{G}$of functions $g : \mathcal{A} \to \mathcal{B}$ is 2-universal (universal for short) if*

$$\forall\, x_1, x_2 \in \mathcal{A}\ \ x_1 \neq x_2 \implies P_G[G(x_1) = G(x_2)] \leq \frac{1}{|\mathcal{B}|}, \tag{2}$$

where G is the random variable that represents the choice of a function $g \in \mathcal{G}$ uniformly at random in $\mathcal{G}$.

In the analysis of this class of systems, collision entropy is shown to be the most suitable information measure, since it better measures the amount of uncertainty faced by Eve regarding the keys agreed upon by Alice and Bob using hash functions from the universal class. Therefore, we will list some important properties of this information measure.

Definition 2. *The collision entropy of a discrete random variable $X \in \mathcal{X}$ is*

$$H_c(X) \triangleq -logE[p_X(x)] = -logP_c(x) = -log\left(\sum_{x\in\mathcal{X}} p_X(x)^2\right), \tag{3}$$

where

$$P_c(x) = \sum_{x\in\mathcal{X}} p_X(x)^2, \tag{4}$$

is collision probability.

For two discrete random variables, $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, the conditional collision entropy of X given Y is

$$H_c(X|Y) \triangleq \sum_{y\in\mathcal{Y}} p_Y(y) H_c(X|Y = y). \tag{5}$$

For any discrete random variables $X \in \mathcal{X}$, the collision entropy satisfies $H(X) \geq H_c(X) \geq 0$. If X is uniformly distributed over $\mathcal{X}$, then $H(X) = H_c(X) = \log|\mathcal{X}|$, where $H(X)$ is Shannon entropy.

The name collision entropy comes from the fact that it is a function of the collision probability (3) of obtaining the same realization of a random variable twice in two independent experiments. For a discrete random variable X, the Renyi entropy of order α is

$$R_\alpha(X) = \frac{1}{1-\alpha} log\left(\sum_{x\in\mathcal{X}} p_X(x)^\alpha\right). \tag{6}$$

Therefore, collision entropy is identical to Renyi entropy of order 2, namely, $H_c(X) = R_2(X)$.

The connection between Renyi entropy and PA based on the universal family of hash functions is formulated in the following theorem [7]:

Theorem 1 . *Let $S \in \{0,1\}^n$ be the random variable that represents the common sequence by Alice and Bob, and letEbe the random variable that represents the total knowledge about S available to Eve. Let e be a particular realization of E. If Alice and Bob know the ECRE2,$R_2(S|E = e)$to be at least some constantc, and if they chooseK = G(S) as their secret key, where G is a hash function chosen uniformly at random from a universal family of hash functions $\mathcal{G} : \{0,1\}^n \to \{0,1\}^k$, then*

$$H(K|G, E = e) \geq k - \frac{2^{k-c}}{ln2} \tag{7}$$

Remark 1. *This result claims that Alice and Bob can generate a shared secret key of length $k < c$, if they know the lower bound c of ECRE2, see Figure 2. Combining (7) and the fact that a binary sequence of length k cannot have a Shannon entropy greater than k, we obtain*

$$k \geq H(K|G, E = e) \geq k - \frac{2^{k-c}}{ln2}. \tag{8}$$

This further means that if Alice and Bob choose the length of the shared secret key

$$k_{GLB\_c}(e) = c - s, \tag{9}$$

where s is the security parameters > 0, the generated keys will differ exponentially small insfrom the maximum entropy sequences, while Eve's total information about that secret key will be exponentially small in s. We will call this PA strategy the global lower bound strategy.

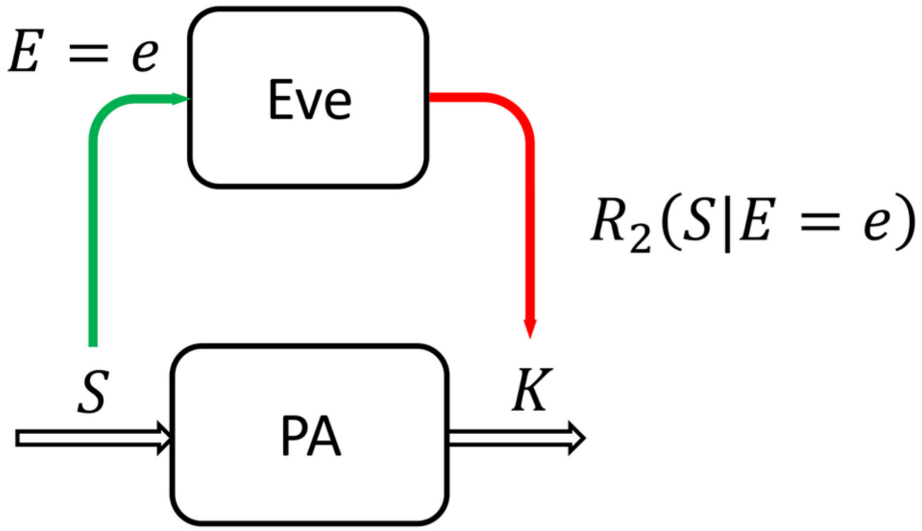

Figure 2. PA with hash functions.

This result dominates today's practice of applying PA in the SKD source model. Since the constant c does not depend on Eve's specific sequence $E = e$, it is clear that for each specific $E = e$, there is a smaller or larger deviation from the established fixed lower bound c, which leads to an unnecessary loss in the length of the generated keys, for the same operating conditions of SKD protocol and the same security parameter $s > 0$.

The following theorem provides the basis for strategy based on the local lower bound for ECRE2.

Theorem 2 . *Let $S \in \{0,1\}^n$ be the random variable that represents the common sequence by Alice and Bob, and let E be the random variable that represents the total knowledge about S available to Eve. Let e be a particular realization of E. If Alice and Bob choose $K = G(S)$ as their secret key, where G is a hash function chosen uniformly at random from a universal family of hash functions $\mathcal{G} : \{0,1\}^n \to \{0,1\}^k$, then*

$$k \geq H(K|G, E = e) \geq k - \frac{2^{k - R_2(S|\ E = e)}}{ln2} \tag{10}$$

Proof of Theorem 2. The proof completely follows the steps of the proof of Theorem 3 from [7], if all relevant probability measures are extended by the additional condition $E = e$. For complete proof see Appendix A. □

In some practical fORapplications, the lower bound for ECRE2 is known to hold with some probability. Theorem 3 answers the question under which conditions in this situation the PA block will ensure the maximum equivocation of the established secret keys from Eve's side.

Theorem 3. *Let $S \in \{0,1\}^n$ be the random variable that represents the common sequence shared by Alice and Bob, and let E be the random variable that represents the total knowledge about S available to Eve. Let e be a particular realization of E. Let the probability that let e be a particular realization of E takes on a value e satisfying $R_2(S|E=e) \geq R_{2\delta}$ is at least $1-\delta$. Let s be an arbitrary security parameter. If Alice and Bob choose $k(e) = R_{2\delta} - s$ as their secret key, where G is a hash function chosen uniformly at random from a universal family of hash functions $G:\{0,1\}^n \to \{0,1\}^k$, then key equivocation from the side of Eve is*

$$H(K|G,E) \geq (1-\delta)\left(k - \frac{1}{ln2}2^{-s}\right). \tag{11}$$

Proof of Theorem 3. By direct application of Theorem 2, we obtain

$$H(K|G,E) = \sum_{all\ e} H(K|G,E=e)p(e) \geq \sum_{all\ e}\left[k - \frac{1}{ln2}2^{k\ -\ R_2(S|E=e)}\right]p(e).$$

Let us divide the set of all sequences e into two sets

$$E_+ = \{e|R_2(S|E=e) \geq R_{2\delta}\}$$

$$E_- = \{e|R_2(S|E=e) < R_{2\delta}\}$$

Then it is valid

$$\begin{aligned}
\sum_{all\ e}\left[k - \tfrac{1}{ln2}2^{k\ -\ R_2(S|E=e)}\right]p(e) &= \sum_{e\in E_-}\left[k - \tfrac{1}{ln2}2^{k\ -\ R_2(S|E=e\ )}\right]p(e) \\
&+ \sum_{e\in E_+}\left[k - \tfrac{1}{ln2}2^{k\ -\ R_2(S|E=e\ )}\right]p(e) \\
&\geq \sum_{e\in E_+}\left[k - \tfrac{1}{ln2}2^{k\ -\ R_2(S|E=e\ )}\right]p(e) \\
&\geq \sum_{e\in E_+}\left[k - \tfrac{1}{ln2}2^{k\ -\ R_{2\delta}}\right]p(e) \\
&= \sum_{e\in E_+}\left[k - \tfrac{1}{ln2}\cdot 2^{-s}\right]p(e) \\
&= (1-\delta)\cdot\left(k - \tfrac{1}{ln2}2^{-s}\right)
\end{aligned}$$

which should have been proved. □

Remark 2. *From Theorem 3, and the fact that the maximum value of Shannon entropy of a binary sequence of length k cannot be greater than k, for a small δ , K has almost maximal entropy for Eve:*

$$k \geq H(K|G,E) \geq (1-\delta)\left(k - \frac{1}{ln2}2^{-s}\right)$$

Remark 3. *If, during the execution of the IR phase of the protocol, n_b parity bits were exchanged over the public channel. According to Lemma 4 [7], it is necessary to perform additional compression for the same amount of n_b bits in the PA phase.*

Based on Theorem 2 and Remarks 1 and 3, we can claim that the *optimal PA strategy* based on the ECRE2 is given by

$$k_{Opt}(e) = R_2(S|E = e) - n_b - s. \tag{12}$$

The main obstacle to the application of this strategy is the fact that ECRE2 is unknown to Alice and Bob since it is conditioned by Eve's sequence e, which is generally unavailable to them. Is it possible to overcome this uncertainty? In this paper, we propose a solution based on machine learning.

4. ML System for Predicting Lower Bound of ECRE2

In the real operating conditions of a given SKD, since the selected DMS is known, it is possible to form training sets of the following structure, see Figure 3:

$$\{[X_{ij}], [Y_{ij}], [Z_{ij}], S_i, e_i\},\ i = 1, \ldots, M,\ j = 1, \ldots, N, \tag{13}$$

where N is the length of individual DMS sequences participating in the protocol, and M is the number of these sequences in the training set. Since S_i, e_i determine $R_{2i}(\ S_i|E = e_i\ )$, we get the training set $\{[X_{ij}], [Y_{ij}], R_{2i}\}$ and its final form

$$\{F_i, R_{2i}\},\ i = 1, \ldots, M. \tag{14}$$

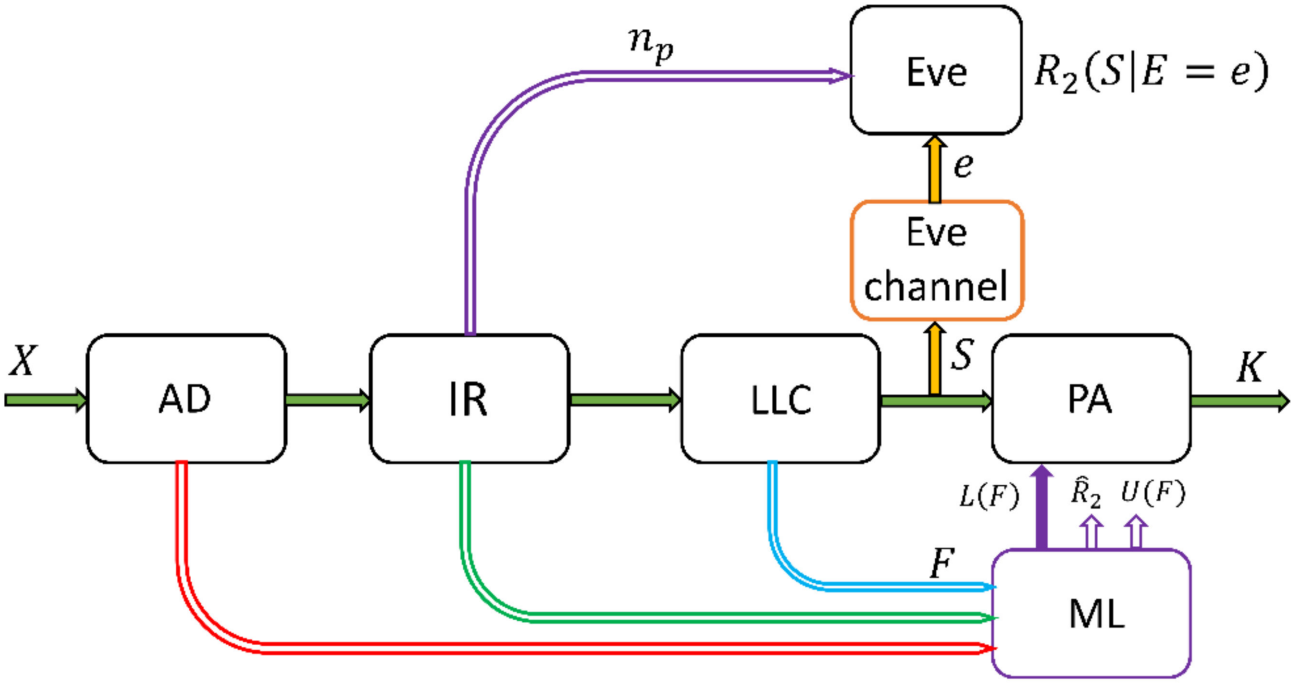

Figure 3. Proposed machine learning system for predicting the lower bound of ECRE2. AD—advantage distillation block, IR—information reconciliation block, LLC—lossless compression block, PA—privacy amplification block, ML—machine learning block.

The feature vectors F_i are formed during the execution of SKD so that they are calculable entirely from the information possessed by Alice. As shown symbolically in Figure 3, these features are usually formed based on information on Alice's side after the execution of individual sub-blocks of SKD. Note that we limit ourselves to the direct SKD, in which Alice starts the protocol and determines the final key length [21]. The same procedure applies to the inverse protocol, in which Alice and Bob switch roles.

Remark 4. *The transition from the training set (13) to the final form (14) requires the calculation of ECRE2 for all pairs* (S_i, e_i). *It can be done based on the expression*

$$R_2(S_i|E = e_i) = -log_2\left(\varepsilon^2 + \left(1 - \varepsilon^2\right)^2\right), \tag{15}$$

where ε *is the bit error probability of equivalent binary symmetric channel (BSC), whose input is* S_i *and output* e_i *[7]. A good estimate of* ε, *is normalized Hamming distance* $D_h(S_i, e_i)$ *between* S_i *and* e_i. *The normalized Hamming distance between two binary sequences X and Y of the same length is given by*

$$D_h(X, Y) = \frac{number\ of\ non\text{-}match\ bits}{number\ of\ bits\ compared}. \tag{16}$$

If the ML block at the output would only provide an estimate $\hat{R}_2$ for ECRE2, which then be used in (12) to calculate the length of the distilled secret key, we have no guarantee that the value $\hat{R}_2$ will be less than the true value of ECRE2. According to Theorem 2, the secret keys generated in this way would not have the desired cryptographic properties of uncertainty and negligible leakage of information to Eve.

Therefore, the ML block should output an interval in which, with a given high probability of $1 - \delta$, the true value of ECRE2 falls, see Figure 4. Then we could use the lower bound $L(F)$ of that interval as an estimate for ECRE2 in (12). According to Theorem 3, we now have a guarantee of the desired cryptographic properties for the obtained keys.

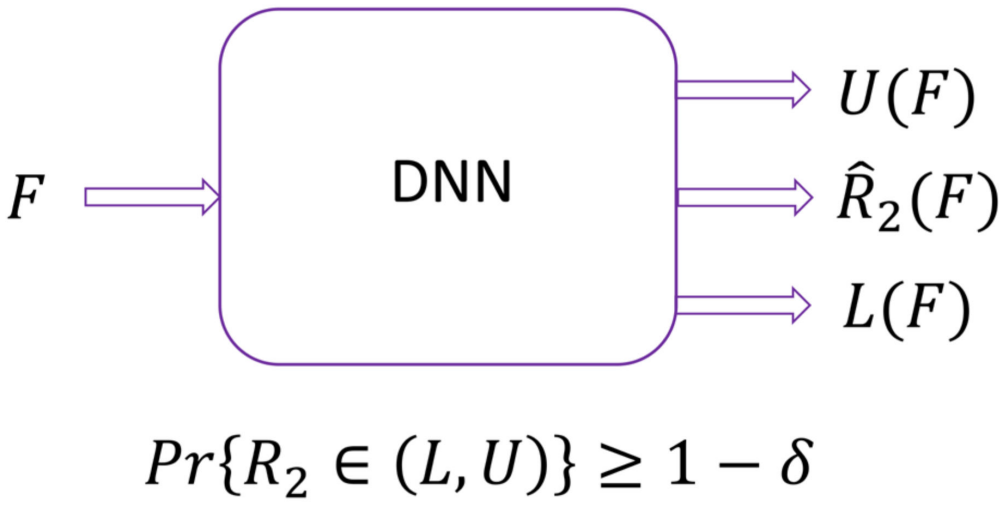

Figure 4. Prediction interval deep neural network (PIDNN) for ECRE2.

In the field of machine learning, the regression block shown in Figure 4 is known as the prediction interval (PI) deep neural network (DNN) model, designed to produce PI for each sample [22–24]. The usual approach to train PIDNN is based on two criterion functions: coverage and mean prediction interval width [25].

Coverage is the ratio of dataset samples that fall within their respective PIs, measured using the prediction interval coverage probability (PICP) metric

$$PICP = \frac{1}{n}\sum_{i=1}^{n} m_i, \tag{17}$$

where n denotes the number of samples and $m_i = 1$ if $R_{2i} \in (L(F_i), U(F_i))$, otherwise $m_i = 0$. It is obvious that *PICP* tends to $1 - \delta$.

Mean prediction interval width (MPIW) is a quality metric for the generated PIs whose goal is producing as tight a bound as possible:

$$PICP = \frac{1}{n} \sum_{i=1}^{n} U(F_i) - L(F_i), \tag{18}$$

The training of PIDNN is performed by the MPIW minimization optimization procedure while keeping the predefined PICP. By combining into a single criterion, we get an unconstrained loss function

$$J_{PI} = MPIW_\theta + \lambda \Psi(1 - \delta - PICP_\theta), \tag{19}$$

$$\Psi(x) = max(0, x)^2, \tag{20}$$

where Ψ is a quadratic penalty function and λ is a hyperparameter controlling the relative importance of width vs. coverage. The algorithm used in this paper is based on the optimization described in [24] and the software package provided on the corresponding GitHub repository [26].

PA strategy based on PIDNN we will call machine learning strategy. It can be formulated by

$$k_{ML}(F_i) = L(F_i) - n_b - s. \tag{21}$$

In the PA system, based on the strategy (21), there may be a situation where $L(F_i) < c$, where c is the global lower bound of ECRE2. Then the global lower bound strategy (9) is better, which justifies the introduction of the next strategy, which we will call the *hybrid PA strategy*

$$k_{Hyb}(F_i) = L_{Hyb}(F_i) - n_b - s. \tag{22}$$

where

$$L_{Hyb}(F_i) = Hyb(L(F_i), R_{2\delta}), \tag{23}$$

while $R_{2\delta}$ is the value of ECRE2 satisfying the condition

$$Prob\{R_2 \geq R_{2\delta}\} \geq 1 - \delta, \ \delta = \frac{1 - PICP}{2}, \tag{24}$$

$PICP$ been coverage value (17) of PIDNN trained on given DMS.

Figure 5 shows the global flow chart of the proposed PA design methodology based on machine learning.

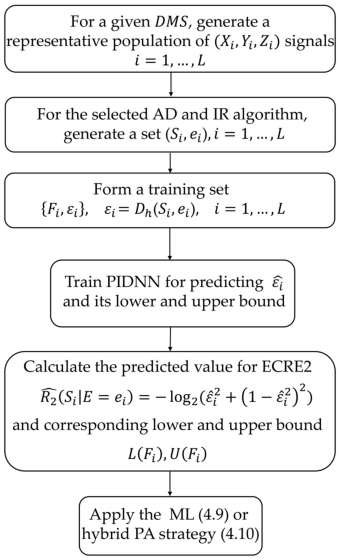

Figure 5. Block diagram of proposed PA strategy.

5. Experimental Evaluation

In the previous section, we defined four different PA strategies:

1. Global lower bound strategy (9);
2. Optimal strategy (12);
3. Machine learning strategy (21); and
4. Hybrid strategy (22).

The experimental evaluation of these strategies was carried out within a general methodological framework, which allows their fair comparison. This implies that different PA strategies are compared by fixing the source, AD and IR part of the SKD, and then varying the PA strategies. In order to get an idea of the dependence of the obtained results on the sources, two different sources were chosen, both in terms of their nature and probabilistic properties.

5.1. Sources of Common Randomness

All evaluation was performed on two sources of common randomness, both obtained from electroencephalography (EEG) signals recorded using the 14-channel EMOTIV EPOC+ wireless EEG headset [27,28]. A detailed argumentation regarding the practical application of this transducer as a source of common randomness in SKD protocols is given in [6].

The first source, which we will call raw EEG, is formed by the serialization of all 14 EEG channel signals. The second source, which we will call EEG metrics, is formed by serialization of 6-dimensional performance metrics, denoted by interest (i.e., emotional valence, attractiveness, or averseness of the task at hand), engagement (or boredom, in negative valence, reflecting the mental workload), excitement (arousal, emotional intensity), stress (frustration), relaxation (meditation), and focus (attention) [28]. Table 1 shows the basic parameters of these sources.

Table 1. This table shows the basic parameters of these sources: sampling rate, length of individual sequences in seconds (length (s)), number of quantization bits per sample (Nbits), as well as the total length of sequences after quantization and serialization (length (bits)).

	Sampling Rate	Sensors/Metrics	Length (s)	Nbits	Length (bits)
Raw EEG	128	14	2	10	35,840
EEG metrics	2	6	300	10	36,000

The signals were recorded from 50 participants aged 20–65 years selected randomly among the employees of Vlatacom Institute of High Technology, Belgrade, Serbia. The participants were aware of the research procedure, including the application of the sensors, and voluntarily agreed to take the test. The institutional ethics committee approved this research following the principles of the Declaration of Helsinki.

The EEG signal recording session lasted 20 min for each participant, who could do whatever they wanted during that time. As a rule, the participants read web content from the Internet, played games, worked on their projects, or meditated. For each participant, two samples of 2 s of recording for the raw EEG source, or two samples of 300 s for the source of EEG metrics, were then randomly selected. In this way, 100 samples were formed for each source.

Figures 6 and 7 show samples of these two sources in the time domain.

5.2. Architectures of Evaluated Systems

Based on the general architecture of the system from Figure 3, three special variants, which we will denote as System A, B, and C, were selected for experimental evaluation.

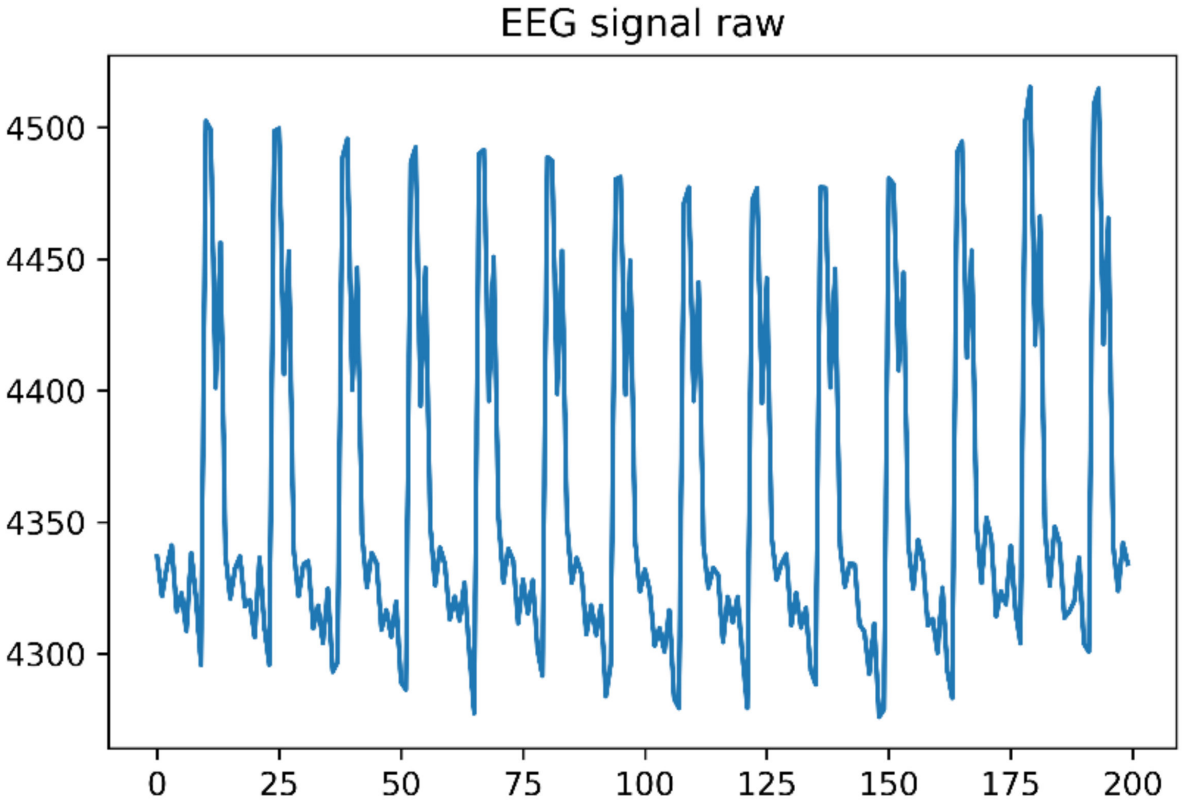

Figure 6. Illustration of the raw EEG source.

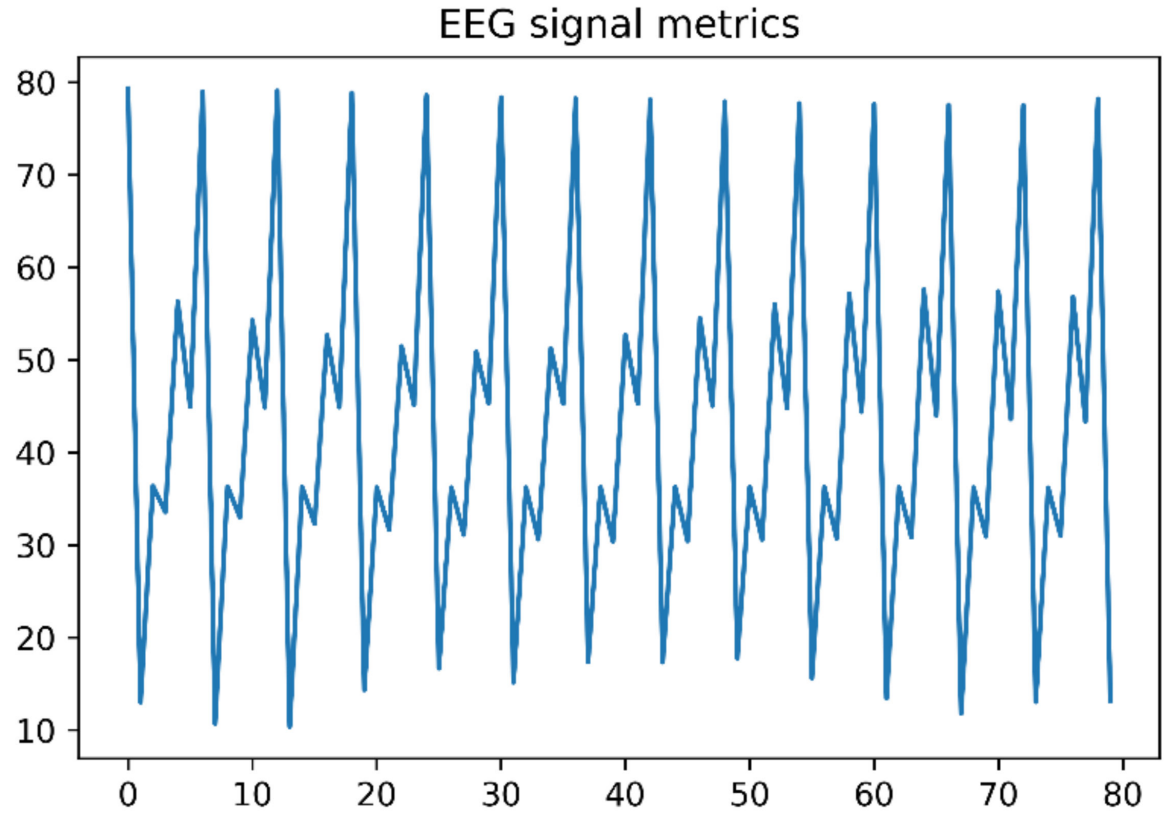

Figure 7. Illustration of the EEG metrics source.

System A consists of the sequence of BP ADD→CASCADE→Universal Hash blocks, where the BP ADD block denotes the bit parity advantage distillation/degeneration protocol [29], and the CASCADE block denotes the one class of IR protocols, first proposed in [30]. This protocol has found wide application in the domain of quantum key distribution and, as such, has been continuously improved and optimized. In this paper, we used an implementation described in [31] and its associated GitHub repository [32].

System B consists of the sequence of ADD→WINNOW→Universal Hash blocks, where the WINNOW block denotes a class of IR protocols based on error-correcting codes [33].

System C consists of a sequence of blocks ADD→WINNOW→Huffman coding→Universal Hash, where the Huffman coding block performs lossless compression based on Huffman source codes [34]. Huffman coders are synthesized based on local sequences at the output of the IR block. Since these sequences are the same for Alice and Bob, the resulting encoders will be the same for them and need not be exchanged over the public channel.

For all three systems, the BP ADD algorithm is executed in the AD block since it proved to be significantly more efficient than the standard bit parity advantage distillation (BP AD) algorithm [35]. Figure 8 illustrates this fact. The top row shows the histograms of mutual normalized Hamming distances of all $100 \cdot 99/2 = 4950$ pairs (Alice, Bob) of sequences for the given initial set of 100 sequences. The bottom row shows a histogram of normalized Hamming distances for the same set of pairs after the end of the AD phase. It can be seen that the BP ADD algorithm more significantly increases the correlation between (Alice, Bob) sequences than the BP AD algorithm. Let us keep in mind that identical sequences have a normalized Hamming distance equal to zero, and uncorrelated sequences have a distance equal to 0.5.

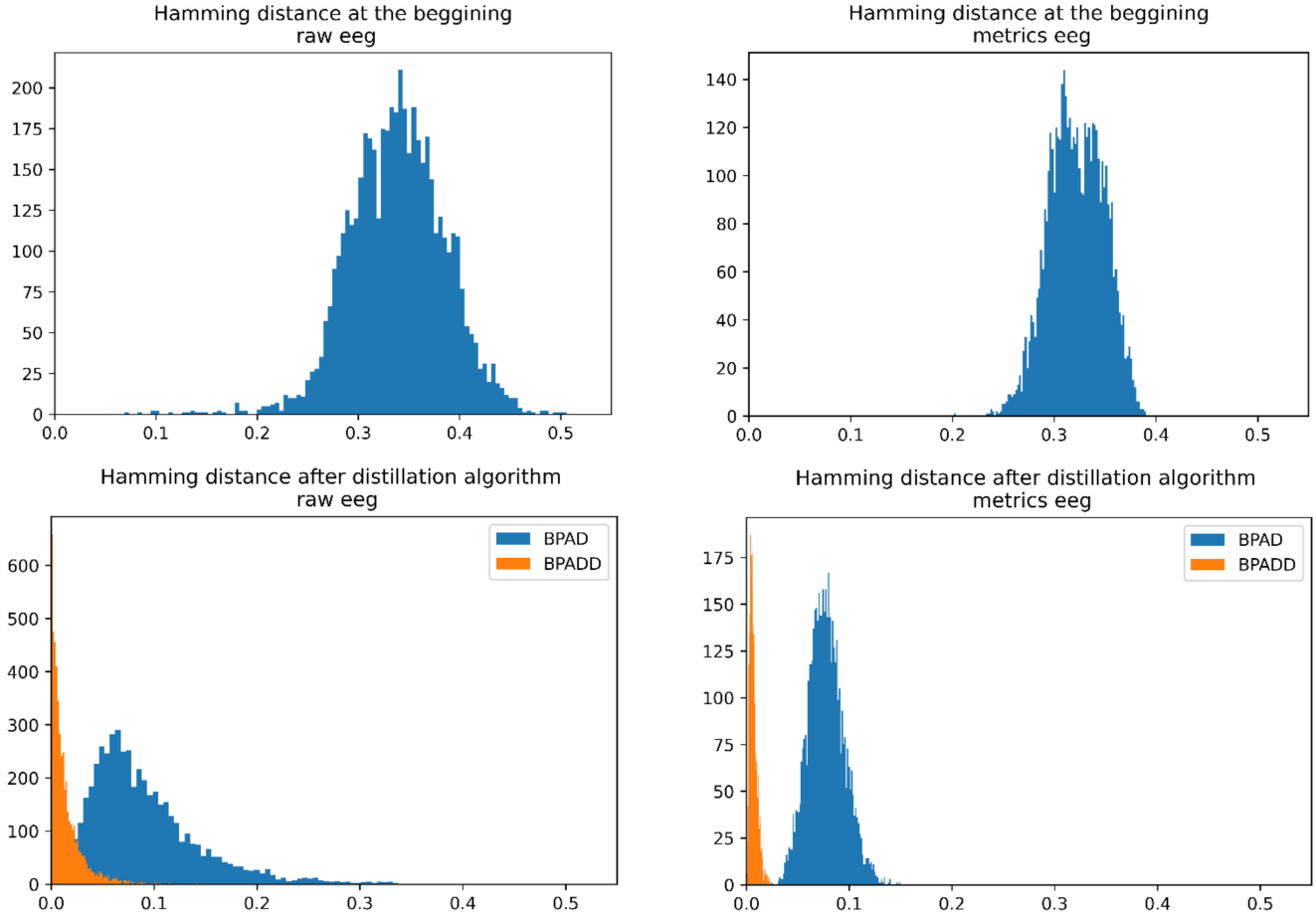

Figure 8. BP AD vs. BP ADD algorithm. It can be seen that the BP ADD algorithm more significantly increases the correlation between (Alice, Bob) sequences than the BP AD algorithm.

The training set is formed for each of the analyzed systems in such a way that for each pair of sequences (Alice, Bob) the corresponding Eve sequence is randomly selected from the remaining sequences. This choice of the eavesdropper is close to the worst-case scenario for the efficiency of the SKD protocol since Eve is actually an insider. Namely, Eve belongs to the population that was selected to represent the future users of the system as representatively as possible.

Table 2 provides an overview of features F, which are measured during the execution of the SKD protocol for all 4950 pairs of legitimate users.

Table 2. Features 10 and 11 were used only for the system with the CASCADE algorithm because, in this information reconciliation algorithm, no security bits were discarded for privacy maintenance. In the case of the Winnow algorithm, security bits are discarded during the algorithm.

Feature No.	Block	Description
1	Start	Length of the sequence at the beginning
2	Start	Length of the sequence after the first iteration of the AD algorithm
3	Start	Length of the sequence after the second iteration of the AD algorithm
4	Start	Normalized block entropy with block size = 8 at the beginning
5	AD	Normalized block entropy with block size = 8 after the first iteration of the AD algorithm
6	AD	Normalized block entropy with block size = 8 after the second iteration of the AD algorithm
7	AD	Number of parity messages exchanged during the AD algorithm
8	IR	Length of the sequence after the IR algorithm
9	IR	Normalized block entropy with block size = 8 after the IR algorithm
10 *	IR	Number of parity messages exchanged during the IR algorithm
11 *	IR	Number of bits that eavesdropper found out during the IR algorithm
12	LLC	Length of the sequence after the lossless compression algorithm
13	LLC	Normalized block entropy with block size = 8 after the lossless compression algorithm

Having in mind Remark 4, the normalized Hamming distance $\varepsilon_i = D_h(S_i, e_i)$ between S_i and e_i was taken for the desired output in the training set. This practically means that PIDNN is trained to predict ε, and then this output is translated into ECRE2 by functional transformation (15). Experimental results show that this approach is more efficient than direct ECRE2 prediction.

Figures 9 and 10 shows architecture for proposed PIDNN systems.

PIDNN training was performed by 10-fold cross-validation over all 4950 pairs of legitimate users. Lower bound predictions L were calculated on the corresponding test sets for each cross-validation iteration. In this way, after completing 10 rounds of cross-validation, predictions were obtained for all 4950 pairs of legitimate users while preserving the independence of the training and test sets. Parameters of learning algorithm are: a number of epochs = 400, batch size = 32, with Adam optimizer and learning rate 0.0005 [28]. For PI algorithm parameters are: λ = 15, and PICP = 0.95. Figure 11 shows loss during training of PINDD.

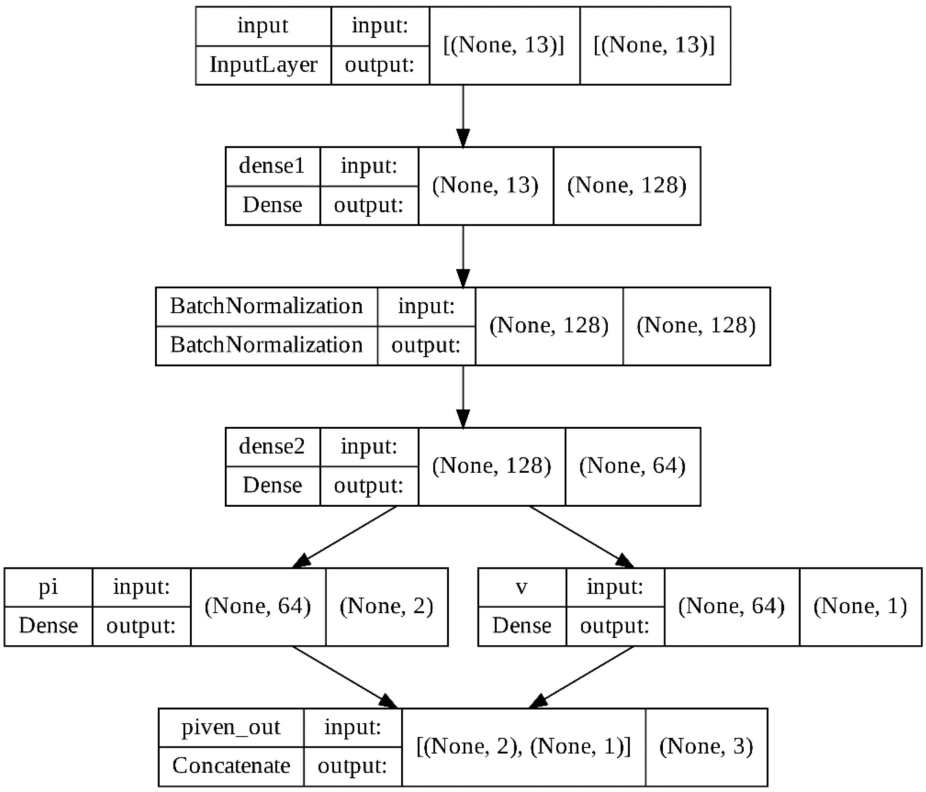

Figure 9. Architecture of PIDNN for System A (obtained from the keras API, [36]).

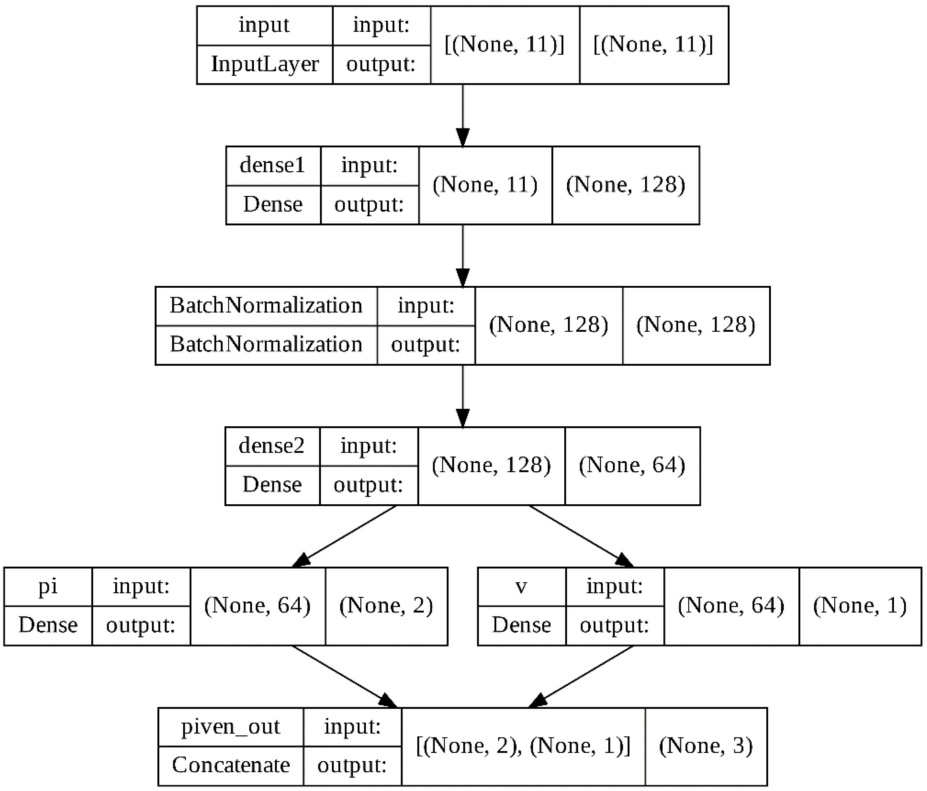

Figure 10. Architecture of PIDNN for System B and C (obtained from the keras API, [36]).

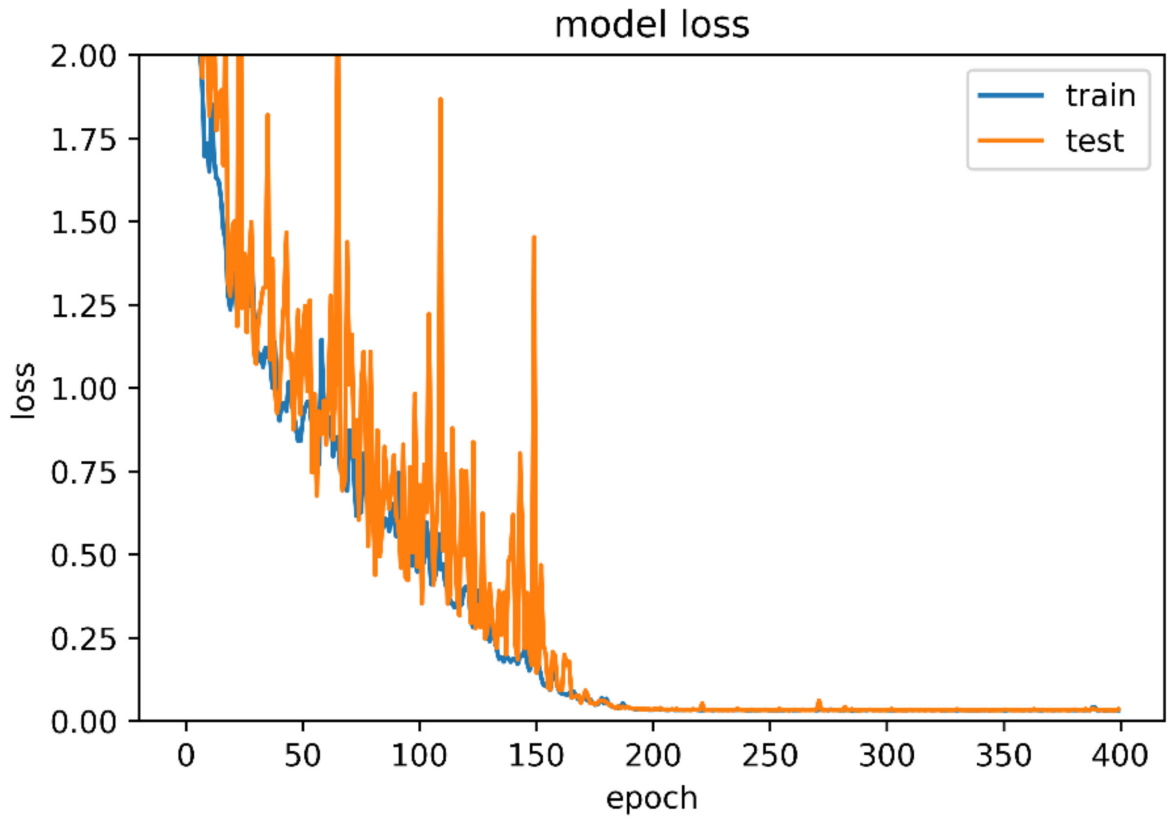

Figure 11. Typical behavior of model loss (19) on training and test set during training of PIDNN.

Figure 12 shows output of PIDNN model.

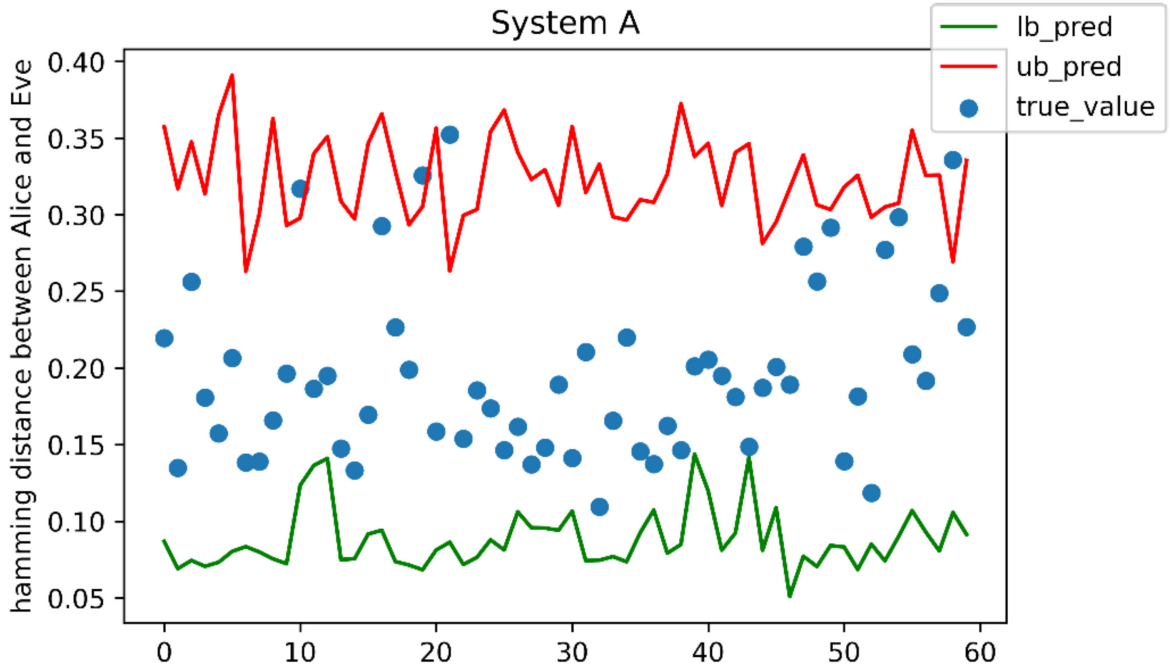

Figure 12. Illustrative sample of output of PIDNN before functional transformation (15).

5.3. Performance Measures

In order to compare individual PA strategies, we will introduce two indicators, gain and loss. Gain of PA strategy A over PA strategy B is defined as follows

$$G_B^A = \frac{|K_A|}{|K_B|}, \tag{25}$$

where $|K_A|$ and $|K_B|$ denote the total length of generated keys using strategies A and B for the same input sequences S. Loss of PA strategy A is defined by

$$Loss_A = \frac{|K_{Opt} - K_A|}{K_{Opt}} \cdot 100\ [\%], \tag{26}$$

where K_{Opt} is the total length of generated keys using optimal PA strategy (12) for the same input sequences S.

The quantities appearing in Tables 3 and 4 are defined as follows. PICP is given by (17), MPIW is defined by (18), R2 denotes the mean value, while $R_{2\min}$ is the minimum value of ECRE2 over the entire population of size L of the given DMS

$$R2 = \frac{1}{L}\sum_{i=1}^{L} R_2(S_i|E = e_i), \tag{27}$$

$$R_{2min} = \min_i R_2(S_i|E = e_i). \tag{28}$$

The mean value of the ECRE2 lower bound L obtained from PIDNN is denoted by

$$R2_{ML} = \frac{1}{L}\sum_{i=1}^{L} L(F_i). \tag{29}$$

while the mean value of the corresponding lower bound L_{Hyb} from the hybrid PA strategy is denoted by

$$R2_{Hyb} = \frac{1}{L}\sum_{i=1}^{L} L_{Hyb}(F_i), \tag{30}$$

$$G_{GLB\_R2min}^{Opt} = \frac{R2}{R_{2min}}, \tag{31}$$

which represents the potential gain in the length of the generated keys when applying the optimal PA strategy over the strategy based on the global lower bound R_{2min}. Similarly,

$$G_{GLB\_R2min}^{ML} = \frac{R2_{ML}}{R_{2min}}, \tag{32}$$

is the gain in the length of the generated keys when applying the PA strategy based on machine learning compared to the standard procedure based on the global lower bound R_{2min} for ECRE2. Corresponding losses

$$Loss_{GLB\_R2min} = \frac{|R2 - R_{2min}|}{R2} \cdot 100\ [\%], \tag{33}$$

$$Loss_{ML} = \frac{|R2 - R2_{ML}|}{R2} \cdot 100\ [\%], \tag{34}$$

express percentage of the unused of a given DMS when applying global lower bound and machine learning PA strategy, respectively.

Table 3. Summary of all indicators for System A, B, and C and both sources: raw EEG and EEG metrics. Columns labeled Mean A, Mean B, and Mean C represent averaging over sources, while the column total mean gives average values over both sources and systems.

	A (Cas-Hash)		**Mean A**	**B (Win-Hash)**		**Mean B**	**C (Win-Huff-Hash)**		**Mean C**	**Total Mean**
	Raw	**Metrics**		**Raw**	**Metrics**		**Raw**	**Metrics**		
$R > 0$ (%)	99.35	100	99.68	99.47	99.94	99.71	100	100	100	99.79
PICP	0.9602	0.9756	0.9679	0.9604	0.9636	0.9620	0.9919	0.9978	0.99485	0.9749
MPIW	0.315	0.221	0.268	0.323	0.215	0.269	0.072	0.063	0.0675	0.2015
R2	2650.71	2609.42	2630.07	2106.40	2131.33	2118.87	2389.59	2993.76	2691.68	2480.20
$R_{2\min}$	319.91	1157.49	738.70	244.93	981.90	613.42	512.88	1612.42	1062.65	804.92
$R2_{ML}$	1064.21	1425.47	1244.84	847.95	1160.49	1004.22	2369.37	2973.56	2671.47	1640.18
$G^{Opt}_{GLB\_R2\min}$	8.2858	2.2544	5.2701	8.6000	2.1706	5.3853	4.6592	1.8567	3.2580	4.6378
$G^{ML}_{GLB\_R2\min}$	3.3266	1.2315	2.2791	3.4620	1.1819	2.3220	4.6197	1.8442	3.2320	2.6110
$Loss_{GLB\_R2\min}$ (%)	87.93	55.64	71.79	88.37	53.93	71.15	78.54	46.14	62.34	68.43
$Loss_{ML}$ (%)	59.82	45.37	52.60	59.66	45.53	52.60	0.85	0.67	0.76	35.32
KR_{ML} (%)	1.12	1.89	1.51	2.36	3.22	2.79	6.61	8.26	7.44	3.91
KAR_{ML} (%)	84.57	99.74	92.16	99.47	99.94	99.71	100	100	100	97.29
LR_{ML} (10^{-3})	4.319	1.487	2.903	1.173	0.684	0.929	0.349	0.241	0.295	1.376
LR_{HLL} (10^{-3})	233.383	290.447	261.915	219.418	282.512	250.965	0.512	0.333	0.423	171.101

Table 4. Randomness test results of the key sequences. The tests are based on the Statistical Test Suite developed by NIST, and results are presented in terms of *p*-values. Initial letters indicate test names: F—frequency, BF—block frequency, R—runs, LR—longest run, FFT—fast Fourier transformation, S—serial, AE—approximate entropy, CSf—cumulative sums forward, CSr—cumulative sums reverse. Tested sequences have a length of 12 million bits for systems A, B, and C, respectively. $p - value\ treshold = 0.01$.

		F	**BF**	**R**	**LR**	**FFT**	**S**	**AE**	**CSf**	**CSr**
A	Raw	0.9167	0.3976	0.3360	0.4400	0.9535	0.0678	0.3218	0.7410	0.8362
	Metrics	0.3399	0.8281	0.4842	0.8010	0.8523	0.0506	0.8992	0.5478	0.2525
B	Raw	0.9276	0.6909	0.5960	0.4376	0.8193	0.5114	0.3720	0.3939	0.4603
	Metrics	0.8308	0.2789	0.8185	0.8228	0.9878	0.1349	0.6255	0.9370	0.9590
C	Raw	0.0748	0.5075	0.3403	0.8959	0.2386	0.3444	0.0995	0.0701	0.1108
	Metrics	0.4152	0.6420	0.6948	0.4182	0.9828	0.0741	0.5117	0.6581	0.5525

The key rate is given by

$$KR = \frac{total\ length\ of\ established\ keys}{total\ length\ of\ input\ sequences} \cdot 100\ [\%], \tag{35}$$

while the key acceptance rate is given by

$$KAR = \frac{number\ of\ final\ keys\ with\ length\ >\ 0}{total\ number\ of\ keys} \cdot 100\ [\%]. \tag{36}$$

The leakage rate measures the amount of information per bit contained in Eve's keys about Alice and Bob's common keys:

$$LR = I(X;Z) = 1\ -\ h_b(D_h(A,E)), \tag{37}$$

where $h_b(x) = -x \cdot \log(x) - (1-x) \cdot \log(1-x)$, $0 < x < 1$, is the binary entropy function.

Quantities (25)–(31) characterize the performance of the system in terms of the length of the generated keys, provided that $s = 0$ and $n_b = 0$. In this way, ECRE2 was highlighted as the dominant factor affecting the lengths of the generated keys. In order to get a clear visual representation of the effects of different strategies on the degree of unused of a given source, specific histograms (see Figure 13) were formed for each of the analyzed systems and sources. First, the ECRE2 histogram of the given source is calculated (blue color). It is also the theoretical limit of the possible length of generated keys, according to (12). Then we keep the bin structure of this basic histogram and calculate over them the histogram of key lengths generated by ML PA strategies (21) (yellow) and global lower bound PA strategy (9) (brown). The areas of these histograms are proportional to the total length of the generated keys of the corresponding PA strategies. Therefore, $R2$ (27) is proportional to the blue area, $R2_{min}$ (28) to the brown area, and $R2_{ML}$ (29) to the yellow area. Gain indicators (31) and (32) are the ratios of blue and brown as well as yellow and brown areas, respectively. Loss indicators (33) and (34) can be interpreted similarly.

Table 3 summarizes all indicators for all three systems and both DMS.

Table 4 shows the results of the randomness tests of all generated key sequences. The randomness tests are based on the Statistical Test Suite developed by the U.S. National Institute of Standards and Technology NIST [15]. The outcome of each experiment is represented by the p-value. An individual test is considered to be passed successfully if the obtained p-value is higher than the threshold of 0.01. Following the obtained results, it can be seen that all generated key sequences meet the defined randomness criteria in all presented tests.

The results of testing the hybrid strategy are given in Table 5. The value of $R_{2\delta}$ is defined by (24), while the other indicators are given by

$$G^{Hyb}_{GLB\_R2min} = \frac{R2_{Hyb}}{R_{2min}}, \tag{38}$$

$$G^{Hyb}_{GLB\_R2\delta} = \frac{R2_{Hyb}}{R_{2\delta}}, \tag{39}$$

$$G^{Hyb}_{ML} = \frac{R2_{Hyb}}{R2_{ML}}, \tag{40}$$

$$Loss_{Hyb} = \frac{\left| R2 - R2_{Hyb} \right|}{R2} \cdot 100\ [\%]. \tag{41}$$

Table 6 shows the results of the randomness tests of all generated key sequences.

Histograms of generated key lengths for optimal, hybrid, and global lower bound $R_{2\delta}$ PA strategies are presented in Figure 14.

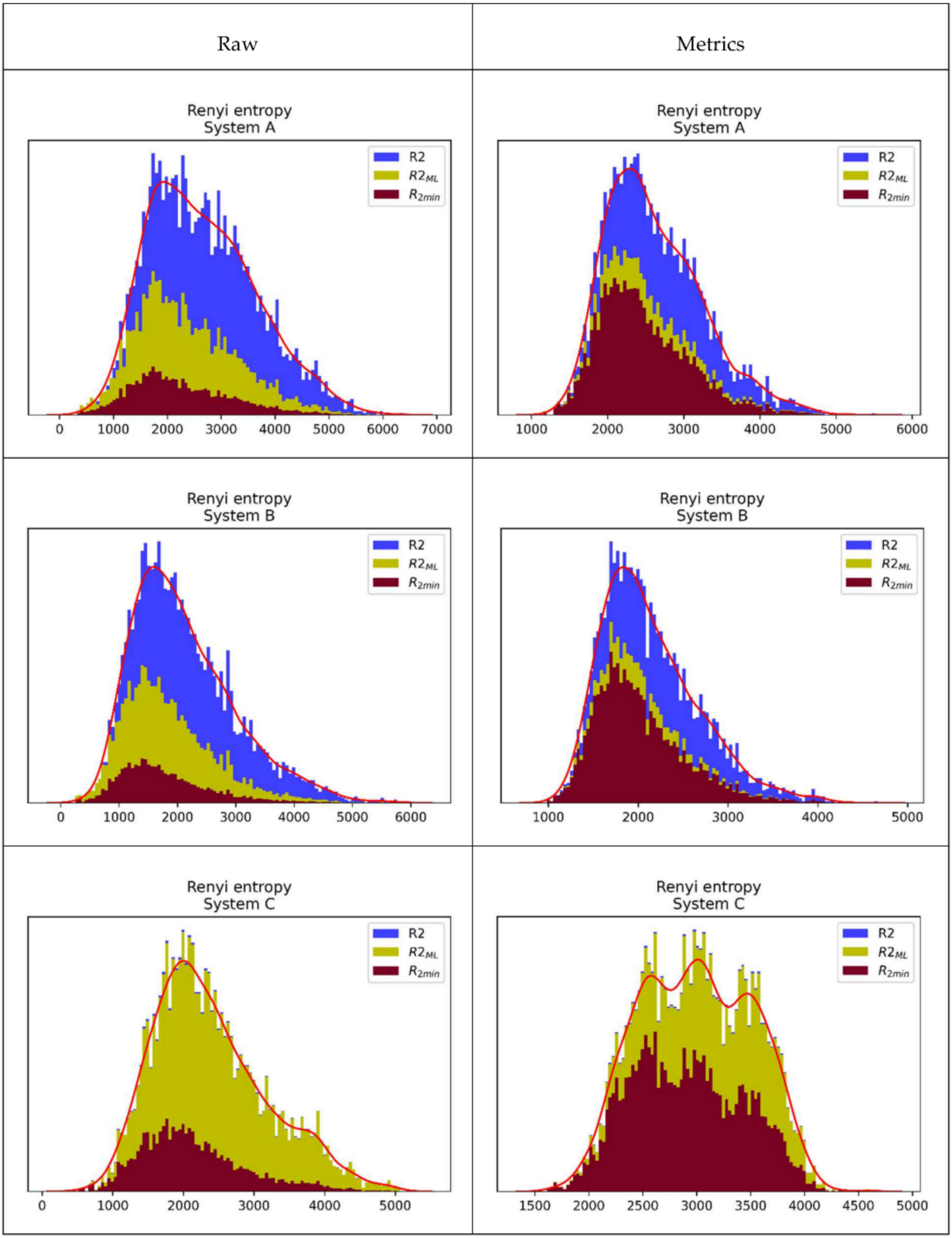

Figure 13. *Blue:* ECRE2 histograms of the sources raw EEG and EEG metrics. It is also a histogram of generated keys by optimal PA strategy (12). *Yellow:* histogram of key lengths generated by ML PA strategies (21). *Brown:* histogram of key lengths generated by global lower bound PA strategy (9). The areas of these histograms are proportional to the total length of the generated keys of the corresponding PA strategies.

Table 5. Summary of all indicators for System A, B, and C and both sources: raw EEG and EEG metrics for hybrid strategies. Columns labeled Mean A, Mean B, and Mean C represent averaging over sources, while the column total mean gives average values over both sources and systems.

	A (cas-hash)		**Mean A**	**B (win-hash)**		**Mean B**	**C (win-huff-hash)**		**Mean C**	**Total Mean**
	Raw	**Metrics**		**Raw**	**Metrics**		**Raw**	**Metrics**		
$R > 0$ [%]	99.35	100	99.68	99.47	99.94	99.71	100	100	100	99.79
PICP	0.9602	0.9756	0.9679	0.9604	0.9636	0.962	0.9919	0.9978	0.9949	0.9749
MPIW	0.315	0.221	0.268	0.323	0.215	0.269	0.072	0.063	0.068	0.2015
$R2$	2650.71	2609.42	2630.07	2106.40	2131.33	2118.87	2389.59	2993.76	2691.68	2480.20
$R_{2\delta}$	1028.65	1555.26	1291.96	816.10	1303.18	1059.64	850.71	1694.77	1272.74	1208.11
$R2_{Hyb}$	1169.26	1584.56	1376.91	928.87	1328.10	1128.49	2369.99	2973.61	2671.80	1725.73
$G^{Hyb}_{GLB\_R2\min}$	3.6550	1.3690	2.5120	3.7925	1.3526	2.5726	4.6210	1.8442	3.2326	2.7724
$G^{Hyb}_{GLB\_R2\delta}$	1.1367	1.0188	1.0778	1.1382	1.0191	1.0787	2.7859	1.7546	2.2703	1.4756
G^{Hyb}_{ML}	1.0987	1.1116	1.1052	1.0954	1.1444	1.1199	1.0003	1.0000	1.0002	1.0751
$Loss_{Hyb}$ (%)	55.89	39.28	47.59	55.90	37.69	46.80	0.82	0.67	0.75	31.71
KR_{Hyb} (%)	1.37	2.33	1.85	2.59	3.69	3.14	6.61	8.26	7.44	4.14
KAR_{Hyb} (%)	93.01	100	96.51	100	100	100	100	100	100	98.84
LR_{Hyb} (10^{-3})	3.106	0.875	1.991	0.840	0.545	0.693	0.343	0.240	0.292	0.992
LR_{HLL} (10^{-3})	233.383	290.447	261.915	219.418	282.512	250.965	0.512	0.333	0.423	171.101

Table 6. Randomness test results of all generated key sequences by the PA Hybrid strategy. The tests are based on the Statistical Test Suite developed by NIST, and results are presented in terms of *p*-values. Initial letters indicate test names: F—frequency, BF—block frequency, R—runs, LR—longest run, FFT—fast Fourier transformation, S—serial, AE—approximate entropy, CSf—cumulative sums forward, CSr—cumulative sums reverse. Tested sequences have a length of 12 million bits for systems A, B, and C, respectively. $p - value\ treshold = 0.01$.

		F	**BF**	**R**	**LR**	**FFT**	**S**	**AE**	**CSf**	**CSr**
A	Raw	0.0732	0.9533	0.3227	0.8138	0.2654	0.1771	0.3879	0.0908	0.0511
	Metrics	0.1287	0.2838	0.2788	0.0775	0.0766	0.4990	0.1135	0.2339	0.0609
B	Raw	0.4059	0.5174	0.8757	0.1129	0.5928	0.0293	0.8818	0.1801	0.7563
	Metrics	0.1742	0.8863	0.0690	0.0819	0.2042	0.3527	0.5038	0.2692	0.3301
C	Raw	0.4794	0.8560	0.1586	0.2438	0.6685	0.7032	0.2597	0.8810	0.4310
	Metrics	0.6468	0.1673	0.6297	0.9099	0.7037	0.3355	0.0257	0.4634	0.8687

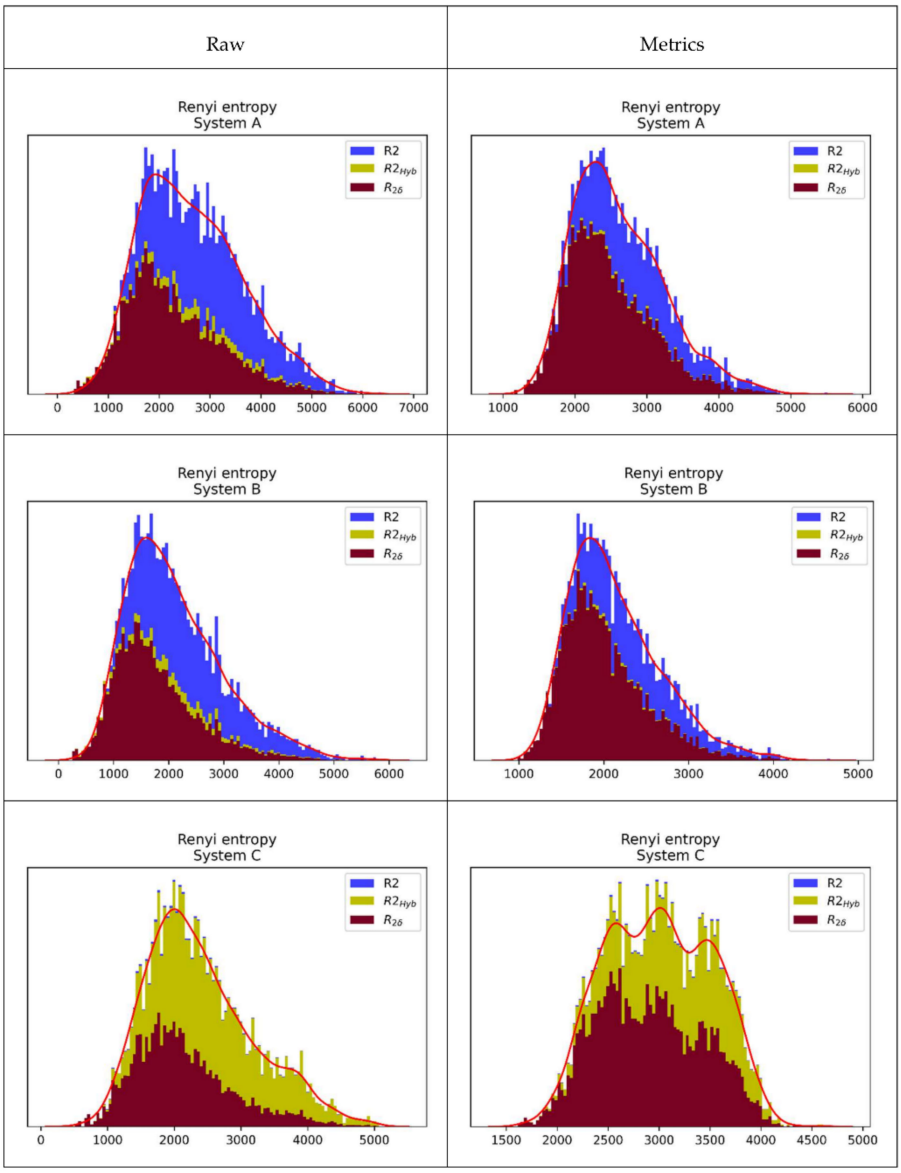

Figure 14. *Blue:* ECRE2 histograms of the sources raw EEG and EEG metrics. It is also a histogram of generated keys by optimal PA strategy (12). *Yellow:* histogram of key lengths generated by hybrid PA strategies (22). *Brown:* histogram of key lengths generated by global lower bound PA strategy with bound $R_{2\delta}$ according to (24). The areas of these histograms are proportional to the total length of the generated keys of the corresponding PA strategies.

Based on all the data presented in Tables 3–6, the following conclusions can be drawn.

(a) For all systems and all sources, the generated keys passed NIST randomness tests with a high margin of confidence;

(b) On average, across all systems and sources, the GLB strategy leaves almost 68% of ECRE2 potential unused. By introducing the ML strategy, the unused is reduced to 35%, and the hybrid strategy would further reduce this indicator to 32%;

(c) On average, across all systems and sources, the ML strategy increases the quantity of generated keys by 2.61 times compared to the GLB strategy. The hybrid strategy further increases this indicator to 2.77 times;

(d) The average value for the δ parameter across all systems and sources is $\frac{1-0.9749}{2} = 0.012$, see (24). Since it is a very small value, according to Remark 2, generated keys have

almost maximal entropy for Eve. The further consequence of this fact is the very small average leakage rate $LR = 1.38 \times 10^{-3}$ that is $LR_{Hyb} = 0.99 \times 10^{-3}$ per one key bit;

(e) Note the dramatic difference of two orders of magnitude in the leakage rate between the classic HLL strategy and the proposed ML hybrid strategy (LR_{HLL} *vs*. LR_{Hyb}, last two rows of Table 5);

(f) On average, across all systems and sources in terms of KR indicators, hybrid strategies give better results than non-hybrid ones (4.41 vs. 3.91). The same is valid for the KAR indicator (98.84% vs. 97.29%);

(g) Averaged by sources, the performance of systems A, B, and C are ranked as A < B < C for all important indicators, such as KR, KAR, LR, and $G^{ML}_{GLB\_R2\min}$. The same relationships are observed for hybrid strategies;

(h) System C has a loss close to 0, unlike systems A and B, where this parameter is 52.6%;

(i) Averaged across sources, ML gain with respect to the GLB strategy amounts to (2.28, 2.32, 3.23) for systems A, B, and C, respectively;

(j) In systems A and B, the hybrid strategy provides improvements of 10–12%, while for system C they do not provide a significant improvement compared to the basic ML strategy;

(k) In terms of all performances, system C significantly exceeds systems A and B by giving $G^{ML}_{GLB\_R2\min} = 3.23$, $Loss_{ML} = 0.76\%$, $LR = 0.295 \times 10^{-3}$, $KAR = 100\%$, and $KR = 7.44$.

5.4. Security Analysis

The presented PA strategies based on machine learning introduce an additional ML block compared to standard PA strategies. From the point of view of security, this does not reduce the uncertainty of the generated keys since the output of this block is the output dimension of the applied hash function, which is also sent to all parties via the public channel as a public parameter in the original version of the protocol.

In the training phase, the system designer can incorporate additional a priori information about the expected Eve strategy. This strategy can be different from the worst-case scenario, which we adopted in this research, taking Eve as a de facto insider from the set of expected users of the system.

The introduction of a block for lossless compression based on Huffman's optimal coding also does not impair the security performance of the system. Namely, the Huffman coder is uniquely determined by the common sequence that Alice and Bob have before the PA block. Therefore, Alice and Bob generate their own Huffman codes locally without requiring any additional communication over the public channel. Eve's strategy, in this case, can be two-fold. The first possibility is to generate a local Huffman coder on its sequence. In that case, any mismatch of its sequence with the Alice, Bob sequence has an effect very similar to applying some equivalent hash function with the same degree of compression. Another possibility is that Eve owns a local Huffman encoder formed by Alice and Bob. Due to the mismatch of the coder with the local Eve sequence, the output sequence will be very similar to the result of the equivalent hash function, as in the first strategy. The experimental evaluation shows that the Huffman coder not only does not compromise the security performance of the system but improves it to a significant extent, which is best seen by the significantly lower value of the LR indicator of system C compared to systems A and B, regardless of the type of source and type of strategy (hybrid or not hybrid).

6. Conclusions

In this paper, a new methodology for the synthesis of the PA block of the SKD system based on machine learning was introduced. In offline mode, before the execution of the protocol itself, the PIDNN was trained on the training set drawn from the given DMS. In protocol execution mode, trained PIDNN gives a local lower bound for ECRE2 with high precision and confidence.

The proposed theoretical-empirical methodology of PA block analysis and design allows us to give new answers to two difficult questions posed in the introductory part of the paper,

1. How much of the total available pure randomness is allocated to secret keys?
2. Is there any leakage toward eavesdroppers, and what is the real security margin of the generated secret keys?

The proposed PA block design methodology allows us to quantify both phenomena mentioned in these questions. In addition, it allows us to precisely quantify the advantage of the proposed hybrid ML strategy over previously known GLB and HLL strategies. The proposed ML and hybrid strategies far surpass GLB and HLL classic PA design in all aspects.

In particular, System C, which consists of the sequence of blocks ADD→WINNOW→ Huffman coding→Universal Hash, in terms of efficient utilization of the given DMS, gives almost ideal results (percentage of unused is 0.75%). This property is of particular importance in today's time of "hunger" for efficient sources and methods of generation and distribution of absolutely secret cryptographic keys.

A particularly interesting and unexpected result is the large leakage of information toward Eve in the classical HLL strategy, which today dominates the practical application of SKD. This result shows that although these systems pass NIST randomness tests, such as in [14], they are susceptible to an efficient dictionary-based attack that allows Eve to reconstruct the Alice and Bob strings before entering the hash block.

Our next research efforts will be focused on trying to replace the entire AD—IR—LLC—PA processing chain with a unique machine learning structure.

Author Contributions: Conceptualization, J.R. and M.M.; methodology, M.M. and B.K.; software, J.R.; validation, J.R., M.M., B.K. and M.J.; formal analysis, M.M.; investigation, J.R.; resources, M.J.; data curation, J.R.; writing—original draft preparation, J.R.; writing—review and editing, M.J.; visualization, J.R.; supervision, M.M.; project administration, J.R. and M.J.; funding acquisition, B.K. All authors have read and agreed to the published version of the manuscript.

Funding: The research is funded by the Vlatacom Institute of High Technologies under project #164 EEG_Keys.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: EEG data of all participants can be obtained upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Proof of Theorem 2. Since $H(K|G, E = e) \geq R_2(K|G, E = e)$, it suffices to establish a lower bound for the $R_2(K|G, E = e)$, i.e., ECRE2. Note that

$$\begin{aligned} R_2(K|G, E = e) &= \sum_{g \in \mathcal{G}} p_G(g) R_2(K|G = g, E = e) \\ &= \sum_{g \in \mathcal{G}} p_G(g) \Big(-log E_{K|G=g,E=e} \Big[p_{K|\mathrm{GE}}(K|g, e) \Big] \Big) \\ &\geq -log \left(\sum_{g \in G} p_G(g) E_{K|G=g,E=e} \Big[p_{K|\mathrm{GE}}(K|g, e) \Big] \right) \end{aligned} \tag{A1}$$

where the last inequality follows from the convexity of the function $x \mapsto -logx$ and Jensen's inequality. Now, let $S_1 \in \{0,1\}^n$ and $S_2 \in \{0,1\}^n$ be two random variables that are

independent of each other and independent of G, which are distributed according to $p_{S|E=e}$. Then,

$$P\left[G(S_1)= G(S_2)|G=g\right] = \sum_{kk\in\{0,1\}^k} p_{G(S)|GE}(kk|g,e)p_{G(S)|GE}(kk|g,e)$$
$$= E_{K|G=g,E=e}\Big[p_{K|GE}(K|g,e)\Big],$$

where $kk \in K$, and we can rewrite inequality (A1) as

$$R_2(K|G,E=e) \geq -logP[G(S_1)= G(S_2)] \tag{A2}$$

We now develop an upper bound for $P\left[G\left(S_1\right)= G(S_2)\right]$. By the law of total probability,

$$\begin{array}{rcl} P[G(S_1)= G(S_2)] & = & P[G(S_1)= G(S_2), S_1= S_2]P[S_1= S_2] \\ & + & P[G(S_1)= G(S_2), S_1 \neq S_2]P[S_1 \neq S_2]. \end{array} \tag{A3}$$

Note that $P[G(S_1)= G(S_2)|E=e, S_1 = S_2] \leq 1$ and $P[S_1 \neq S_2|E=e] \leq 1$. In addition, by virtue of the definition of collision entropy,

$$P[S_1= S_2] = \sum_{s\in\{0,1\}^n} p_{S|E=e}(s|e)^2 = 2^{-R_2(S|E=e)}.$$

Finally, because the hash function $\mathcal{G}$ is chosen in a universal family, it holds that

$$P[G(S_1)= G(S_2)|S_1 \neq S_2] \leq 2^{-k}.$$

On substituting these inequalities into (A3), we obtain

$$P[G(S_1)= G(S_2)] \leq 2^{-R_2(S|E=e)}+2^{-k} = 2^{-k}\left(1+2^{k-R_2(S|E=e)}\right). \tag{A4}$$

On substituting (A4) into (A2) and using the fact that $ln(1+x) \leq x$ for all $x > -1$, we obtain

$$R_2(K|G,E=e) \geq k - \frac{2^{k-R_2(S|E=e)}}{ln2}$$

Finally, having in mind the lower bound for Shannon entropy, we obtain

$$k \geq H(K|G,E=e) \geq k - \frac{2^{k-R_2(S|E=e)}}{ln2} \quad \square$$

References

1. Shannon, C.E. Communication Theory of Secrecy Systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
2. Wolf, S. Unconditional Security in Cryptography. In *Lectures on Data Security: Modern Cryptology in Theory and Practice, Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1561, pp. 217–250.
3. Ahlswede, R.; Csiszar, I. Common randomness in information theory and cryptography, Part I: Secret sharing. *IEEE Trans. Inf. Theory* **1993**, *39*, 1121–1132. [CrossRef]
4. Maurer, U.M. Secret key agreement by public discussion from common information. *IEEE Trans. Inf. Theory* **1993**, *39*, 733–742. [CrossRef]
5. Csiszar, I.; Narayan, P. Secrecy Capacities for Multiple Terminals. *IEEE Trans. Inf. Theory* **2004**, *50*, 3047–3061. [CrossRef]
6. Galis, M.; Milosavljević, M.; Jevremović, A.; Banjac, Z.; Makarov, A.; Radomirović, J. Secret-Key Agreement by Asynchronous EEG over Authenticated Public Channels. *Entropy* **2021**, *23*, 1327. [CrossRef] [PubMed]
7. Bennett, C.H.; Brassard, G.; Crepeau, C.; Maurer, U.M. Generalized privacy amplification. *IEEE Trans. Inf. Theory* **1995**, *41*, 1915–1923. [CrossRef]
8. Bennett, C.H.; Brassard, G.; Robert, J.M. How to reduce your enemy's information. In *Advances in Cryptology, Proceedings of Crypto '85, Lecture Notes in Computer Science*; Springer: Berlin, Germany, 1986; Volume 218, pp. 468–476.

9. Bennett, C.H.; Brassard, G.; Robert, J.-M. Privacy Amplification by Public Discussion. *SIAM J. Comput.* **1988**, *17*, 210–229. [CrossRef]
10. Zhang, J.; Duong, T.Q.; Marshall, A.; Woods, R. Key Generation from Wireless Channels: A Review. *IEEE Access* **2016**, *4*, 614–626. [CrossRef]
11. Impagliazzo, R.; Levin, L.A.; Michael, L. Pseudo-random Generation from one-way functions. In Proceedings of the 21st Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, 14–17 May 1989; pp. 12–24.
12. Guglielmi, A.V.; Muraro, A.; Cisotto, G.; Laurenti, N. Information Theoretic Key Agreement Protocol based on ECG signals. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; pp. 1–6. [CrossRef]
13. Xu, W.; Javali, C.; Revadigar, G.; Luo, C.; Bergmann, N.; Hu, W. Gait-Key: A gait-based shared secret key generation protocol for wearable devices. *ACM Trans. Sen. Netw.* **2017**, *13*, 1–27. [CrossRef]
14. Li, G.; Hu, A.; Zhang, J.; Peng, L.; Sun, C.; Cao, D. High-Agreement Uncorrelated Secret Key Generation Based on Principal Component Analysis Preprocessing. *IEEE Trans. Commun.* **2018**, *66*, 3022–3034. [CrossRef]
15. NIST. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Commerce: Gaithersburg, MD, USA, 2010. Available online: https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final (accessed on 6 October 2021).
16. Wu, J.; Guo, S.; Huang, H.; Liu, W.; Xiang, Y. Information and Communications Technologies for Sustainable Development Goals: State-of-the-Art, Needs and Perspectives. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2389–2406. [CrossRef]
17. Ma, B.; Wu, J.; Lai, E.; Hu, S. PPDTSA: Privacy-preserving Deep Transformation Self-attention Framework For Object Detection. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; pp. 1–5. [CrossRef]
18. Maurer, U.M.; Wolf, S. Secret-key agreement over unauthenticated public channels—Part III. Privacy amplification. *IEEE Trans. Inf. Theory* **2003**, *49*, 839–885. [CrossRef]
19. Yakovlev, V.; Korzhik, V.I.; Morales-Luna, G.B.; Bakaev, M. Key distribution protocols based on extractors under the condition of noisy channels in the presence of an active adversary. *arXiv* **2010**, arXiv:1005.3184.
20. Carter, J.L.; Wegman, M.N. Universal classes of hash functions. *J. Comput. Syst. Sci.* **1979**, *18*, 143–154. [CrossRef]
21. Bloch, M.; Barros, J. *Physical-Layer Security: From Information Theory to Security Engineering*; Cambridge University Press: Cambridge, UK, 2011.
22. Keren, G.; Cummins, N.; Schuller, B. Calibrated Prediction Intervals for Neural Network Regressors. *IEEE Access* **2018**, *6*, 54033–54041. [CrossRef]
23. Kivaranovic, D.; Johnson, K.; Leeb, H. Adaptive, Distribution/Free Prediction Intervals for Deep Networks. In Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), Palermo, Italy, 26–28 August 2020.
24. Simhayev, E.; Katz, G.; Rokach, L. PIVEN: A Deep Neural Network for Prediction Intervals with Specific Value Prediction. *arXiv* **2020**, arXiv:2006.05139.
25. Khosravi, A.; Nahavandi, S.; Creighton, D.; Atiya, A.F. Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals. *IEEE Trans. Neural Netw.* **2010**, *22*, 337–346. [CrossRef] [PubMed]
26. Elisim/PIVEN-GitHub. Available online: https://github.com/elisim/piven (accessed on 6 August 2022).
27. Emotiv | Brain Data Measuring Hardware and Software Solutions. Available online: http://emotiv.com (accessed on 6 October 2021).
28. Emotiv | EpocX. Available online: https://www.emotiv.com/epoc-x/ (accessed on 6 August 2022).
29. Wang, Q.; Wang, X.; Lv, Q.; Ye, X.; Luo, Y.; You, L. Analysis of the information theoretically secret key agreement by public discussion. *Secur. Commun. Netw.* **2015**, *8*, 2507–2523. [CrossRef]
30. Brassard, G.; Salvail, L. Secret key reconciliation by public discussion. In *Advances in Cryptology—EUROCRYPT'93*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1994; Volume 765, pp. 410–423.
31. Reis, A. Quantum Key Distribution Post Processing—A Study on the Information Reconciliation Cascade Protocol. Master's Thesis, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal, 2019.
32. Brunorijsman/Cascade-Python—GitHub. Available online: https://github.com/brunorijsman/cascade-python (accessed on 6 October 2021).
33. Buttler, W.T.; Lamoreaux, S.K.; Torgerson, J.R.; Nickel, G.H.; Donahue, C.H.; Peterson, C.G. Fast, efficient error reconciliation for quantum cryptography. *Phys. Rev. A* **2003**, *67*, 052303. [CrossRef]
34. Huffman, D.A. A Method for the Construction of Minimum-Redundancy Codes. *Proc. IRE* **1952**, *40*, 1098–1101. [CrossRef]
35. Gander, M.J.; Maurer, U.M. On the secret key rate of binary random variables. In Proceedings of the 1994 International Symposium on Information Theory and Its applications, Sydney, Australia, 20–24 November 1994.
36. Keras API. Available online: https://keras.io/api/utils/model_plotting_utils/#plot_model-function (accessed on 1 May 2022).