Development of a Neural Net-Based, Personalized Secure Communication Link

Dirk Neumann[1], Rolf Eckmiller[1], and Oliver Baruth[1]

Department of Computer Science, Division of Neural Computation, University of Bonn, Römerstr. 164, 53117 Bonn, Germany,
{neumann, eckmiller, baruth}@nero.uni-bonn.de,
http://www.nero.uni-bonn.de

Abstract. This paper describes a novel ultra-secure, unidirectional communication channel for use in public communication networks, which is based on

a) learning algorithms in combination with neural nets for fabrication of a unique pair of modules for encryption and decryption, and
b) in combination with decision trees for the decryption process,
c) signal transformation from spatial to temporal patterns by means of ambiguous spatial-temporal filters (ST filters),
d) absence of public- or private keys, and
e) requirement of biometric data of one of the users for both generation of the pair of hardware/software modules and for the decryption by the receiver.

To achieve these features we have implemented an encryption-unit (EU) using ST filters for encryption and a decryption unit (DU) using learning algorithms and decision trees for decryption.

1 Introduction

To establish a secure communication via a public network (e.g. www) several methods like VPN or SSH are known. All of these methods use a private key to ensure private communication [9], [13]. With the willingness of everyone to use public networks even for very private activities, e.g. information exchange with the medic institute or transactions with your credit institute via the www, the demand on easy to use encryption systems with integrated authentication arises. To enhance the acceptance of a new secure communication environment, the user should not keep an additional password in mind, but rather use his always available biometric identification [5], [15]. The proposed secure communication environment uses biometric data of an owner to establish an individually secured, encrypted point-to-point communication [1], [6]. To achieve this ultra secure communication several different modules have to be developed, some running in software on a normal PC, others being realized on special hardware (e.g. PCMCIA FPGA Card) [18], [17].

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 340–348, 2006.

2 Generation of a Unique Pair of Hardware/Software Encryption- and Decryption Units

The secure communication environment consists of a software component, two hardware components and one biometric input device, e.g. fingertip sensor (see Fig. 1) [11]. Initally, the functionality of the system parts is not given.

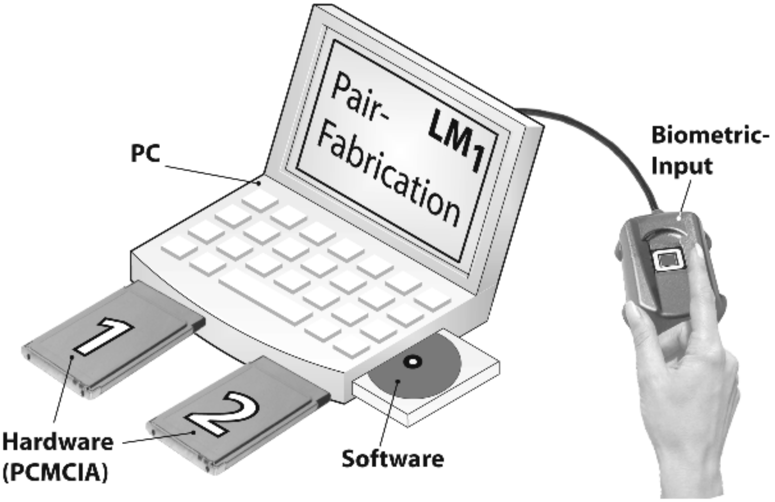

Fig. 1. System components of the secure communication environment. Two special hardware modules (preferably as programmable FPGA PCMCIA cards, depicted as 1 and 2) are programmed by a computer. A special software is used to allow biometric data to be the input to a learning module (LM) whereas the biometric data is acquired via a fingerprint sensor. The learning module LM_1 is used in a interative tuning procedure to generate two different complementary algorithms which will be embedded in the hardware modules forming a unique pair of functional complementary modules.

In a first step, the owner of the hardware kit uses a normal PC to manufacture a unique pair of hardware modules (e.g. PCMCIA FPGA Boards, intelligent smart cards) [3]. Later, these cards are essential to build the encryption unit (EU) and the decryption unit (DU). In this step the biometric data of an owner is acquired via a biometric input device. This data is used as an individual modulation vector (MV) which defines the parameter of the algorithm, of the used spatial temporal filters (ST filter) in the EU.

These filter algorithms will be implemented in hardware on a special module. Additional to the modulation vector a random part (random vector RV) exists like the salt value for hash calculations [9] so that an owner will never generate two identical encryption units. RV is not used for the ST filter algorithm, but rather to deform shape and geometrical arrangement of the original concentric symmetric detection zone (DZ) of the ST filters.

In a second step the learning module one (LM_1) is used to train iteratively a neural network (MLP) in order to define the geometrical arrangement by RV back to original concentric symmetric DZ [10], [14]. For security reasons the function of the neural net will be implemented in the second hardware module, which is used in the decryption unit.

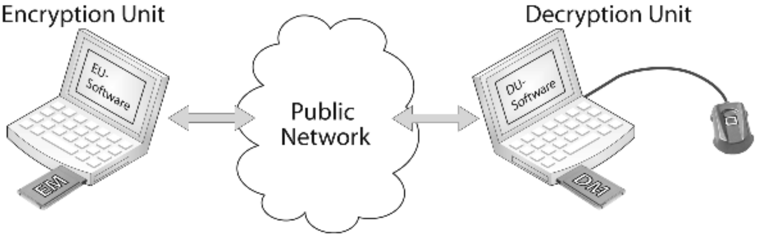

Fig. 2. Left part: A personal computer uses the first hardware module (EM) with its embedded encryption functions. On this hardware module an additional software component is placed in a flash-memory subunit which is used by the PC to act as encryption unit (EU). Right part: A second PC uses the second hardware module (DM) including another software component to form the decryption unit (DU). These two units can use a public network for communication purposes.

An other approach to generate two complementary functions for DU and EU uses only very view (e.g. only one, or defined by RV) ST filters in the encryption unit (EU) which are moved along a trajectory defined by the random vector RV. In this the case the complete input area has to be covered at least two times. In this case the learning module (LM_1) is used to find a representation of this trajectory which can be used by the decryption unit (DU).

The second hardware module also includes a special learned decision tree [10] which uses the ST filter output functions as input and reconstructs the original input data. This decision tree has to know about the used ST filter properties in EU, which are defined by MV. Whereas MV will be acquired via the biometric input device [12]. The required biometric input data (MV) during the decryption process prevent DU from unprohibited usage.

After these two steps the two hardware modules can be used in different locations. The first hardware module connected to a PC forms the encryption unit and the second hardware module connected to an other PC with a biometric input device forms the decryption unit.

3 Encryption Specifics

The encryption unit (EU) is based on the usage of different ST filters. These ST filters have their own assigned detection zone (DZ), which is used to acquire data as a binary input. This binary data can be taken from a black and white picture. Each ST filter does its own data processing (the calculation can be scheduled in parallel) on the acquired data; this computation uses two different independent calculation routines, one for the center and one for the surrounding [2], [4]. The center pixels are indicated as C and the surrounding pixels as S (see Fig. 3).

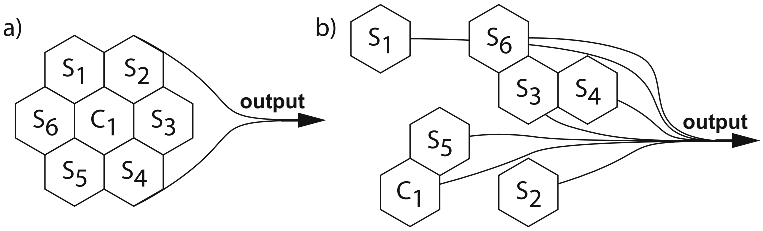

Fig. 3. (a) Example of a simple ST filter with concentric symmetric detection zone (DZ) to show the functionality. This DZ uses one input value for the center calculation (indicated as C_1) and six input values for the surrounding calculation (indicated as S_1, ..., S_6). (b) ST filter with a more spreading DZ which arises from the concentric symmetric DZ shown in (a) by a transformation depending on a random vector (RV).

Each routine uses a weighting (indicated as +, −) of the binary data and then calculates the sum over the corresponding center- and surrounding values. The temporal properties of each ST filter are implemented by means of FIR filters [7]. These ST filters are constructed in such a way that several input patterns will generate the same output value. This gives us a simple 'one way function' [9] because if we only have the output of one ST filter, we cannot calculate the original input pattern. This means that an ST filter always creates ambiguity and we cannot clearly dissolve the ambiguity of an individual ST filter. Now, if we use a special set of ST filters which have to fulfill defined conditions, we can clearly reconstruct the original input pattern [8], [16]. This reconstruction can only be done if all properties of the used ST filters are known. This means that our modulation vector (MV) is a very essential part for this calculation. These ST filters are chosen among a defined set of ST filter classes. These filter classes are defined by MV, and each ST filter is assigned to one class by MV. Each ST filter class has its own properties concerning the detection zone and the time response.

Additional to these requirements we have to define conditions for the arrangement of the detection zones used by the ST filters. The ST filters must be arranged over the whole input pattern, so that each pixel is covered of at least two detections zones of different ST filters. If we have ST filters with only one center pixel, then this center pixel needs not to be covered by an additional ST filter. For example, if we use three different classes of ST filters and each filter class has one center pixel and six surrounding pixels, we get a placement shown in Fig. 5a. We see that the surrounding pixels in the middle of the three ST filters, are each covered by two adjacent ST filters.

If we have more ST filters in a pixel plane and if we have three different ST filter classes, we could assign them to the used filters in a fashion shown in Fig. 5b (this example tiling we call basic tiling). This assignment is defined by MV. To make the encryption unit unique and independently from the used biometric data we use the random vector (RV) to deform the shape and to change the spread of each used ST filter (see Fig. 3b). Whereas these two vectors are embedded in the algorithm, which are embedded in the hardware boards and calculate the ST filter functions.

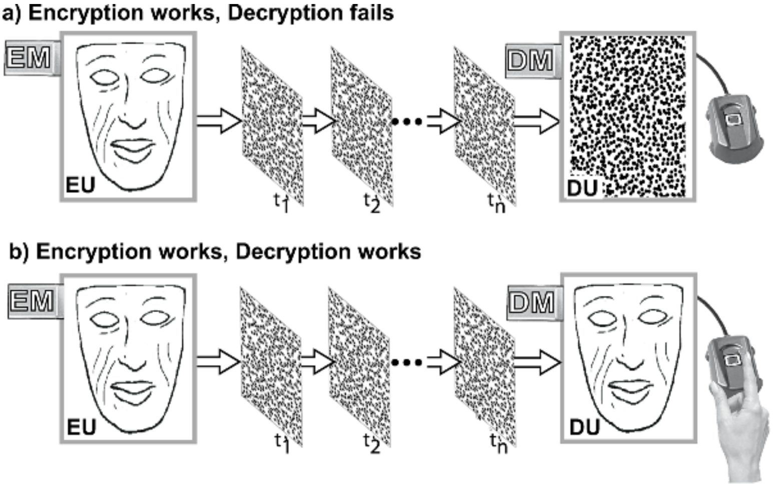
a) Encryption works, Decryption fails
EM
EU
t_1
t_2
t_n
DM
DU
b) Encryption works, Decryption works
EM
EU
t_1
t_2
t_n
DM
DU

Fig. 4. The encryption unit (EU) uses the black and white picture showing a picture of a Inca mask as input. The spatial information of this picture is transformed into a pure temporal data stream by means of FIR filters, depicted by $t_1, \ldots, t_n$. This data stream is received by the corresponding decryption unit (DU) which reconstruct the original picture/spatial information by using of additional information extracted from the biometric input device. Upper part: EU uses the embedded algorithms of the encryption module (EM) to calculate the temporal data stream. The reconstruction of the original data stream fails because of the missing biometric input, and the decrypted image only shows a random image. Bottom part: The EU encrypts the data as described above. The DU is now able to reconstruct the original data, depicted as the Inca mask, with the additional biometric information acquired via the biometric input device.

4 Decryption Specifics

The decryption unit (DU) uses the second hardware board and a biometric input device to reconstruct the original data out of the encrypted data stream. Thereby the hardware board is used to reconstruct the original arrangement of each ST filter detection zone. The software component including a second learning module (LM_2) uses the biometric input device to acquire the modulation vector (MV) and uses it as additional input for a special decision tree (DT). The DT will help to reconstruct the input pattern from the beginning. The biometric input is an essential part of the decryption unit which gives us the possibility to use DU only if we are the owner and only if we permitted to do that.

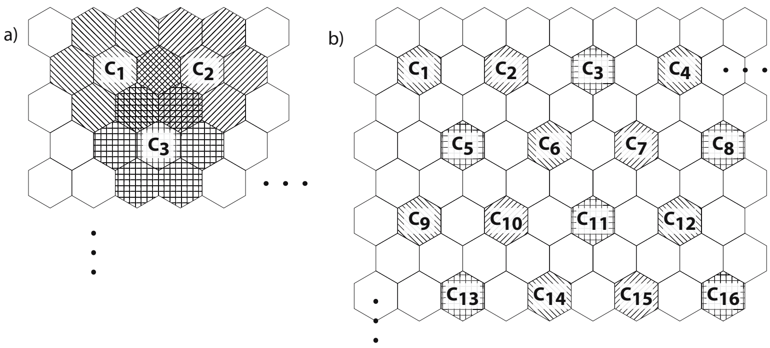

Fig. 5. (a) Example of an ST filter placement showing three different ST filter classes. Each surrounding pixel is covered by at least two different ST filters. The center values are only covered by one ST filter, thereby C_i indicates the center of ST filter i (b) Possible assignment of ST filter to three available ST filter classes over the entire input, fulfilling the properties mentioned for (a). For simplicity reasons only the center values of each ST filter are indicated.

In some cases the information of the ST filters are not sufficient to recalculate the input pattern clearly, so there is an additional part in the software solution which will request an additional set of ST filter outputs. This additional ST filter output is the result of a second calculation of the ST filter, where the position of all ST filters is shifted by one pixel (one bit) in a direction which is specified by DU. In this configuration the decryption unit with its unique second hardware board can only decrypt data which was generated by the corresponding encryption unit, using the corresponding unique first hardware board derived from the initial tuning/manufacture process described above. This means that we have authentication functionality within this unique pair of hardware boards working in the EU and the DU. Somebody can authenticate himself by using the EU to send an encrypted message to the DU, because the DU is only capable to decrypt messages from the corresponding EU.

5 First Simulation Study

To show the feasibility we have implemented a basic version of this proposed secure communication environment. This basic version (see Fig. 6) can be used to test various parameters of the used algorithms. Our implemented encryption unit uses a black and white picture with a resolution of 32x32 pixels as input data (P_{In}). For the used ST filters we developed three different classes, each with the same geometric expansion of the detection zone. The parameter of the weighting and the parameter for the time response were different. For the distribution of the ST filters we used the basic tiling (see Fig. 5b).

To cover the complete input pattern with ST filters it is necessary to use 17x17 of them, each chosen from the three ST filter classes. To transmit the

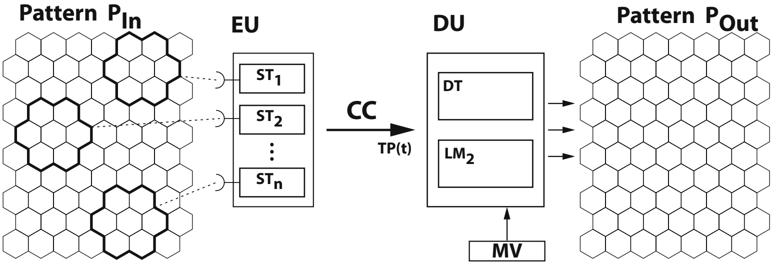

Fig. 6. Schema of the secure communication environment showing the input pattern P_1 scanned by the ST filter ensemble implemented in the encryption unit (EU); this ensemble consists of n ST filters. The calculated temporal data stream output TP(t) of the DU is transmitted via a communication channel CC to the decryption unit (DU). DU use TP(t) with external information obtained via the modulation vector (MV) to use the second learning module (LM_2) and a decision tree (DT) to reconstruct the output pattern P_2 equal to P_1.

output of these ST filters we can use the standard TCP/IP protocol or we can save the output in a file. For the decryption unit we have implemented a transceiver to gather the transmitted encrypted data. This data gets into the initially trained neural net which arranges the detection zones for the postprocessing. The following decision uses the modulation vector to get information of the used ST filters (defined ST filter classes and assignment of each ST filters to one of them). As a result of this learned decision tree we get a 32x32 pixel black and white picture P_{Out} clearly identical to the input picture (see Fig. 7). The implemented decision tree needs, only one additional output from the ST filters to reconstruct the picture, where the filters are shifted by one pixel. The runtime of the decision tree is independent of the used ST filter to ST filter class assignment and independent of the input pattern.

We have shown that single ST filters which produce ambiguity and thus are not clearly invertible can be used in ensembles to allow us to reconstruct the original input. Therefor the used detection zones must fulfill several requirements. With the currently implemented three different ST filter classes, a pair of encryption- (EU) and decryption unit (DU) could be created using the random vector (RV) to get a unique EU-DU pair. This EU-DU pair has an embedded authentication feature and can be used for authentication purposes. The modulation vector (MV) can be used to establish a usage of the DU only if the owner uses its own fingerprint to permit this. For future steps we will implement more types of ST filter classes with differently shaped detection zones. Furthermore these detection zones will overall use more pixels for their calculations.

Another meaningful extension could be a bidirectional communication between two geographically spread locations. For this purpose we have to integrate an encryption- and a decryption unit in one system or better in one hardware board, where encryption- and decryption algorithms are placed in one system. In

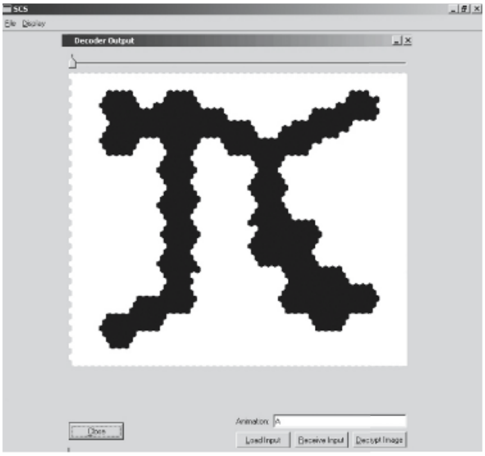

Fig. 7. Screen shot of the working decryption unit (DU). It uses the information from the modulation vector (MV) for decryption and finally shows the window of the decrypted image, in this case a black and white picture of π.

this case, encryption and decryption should have different modulation vectors. Thereby we have two different pairs of EU-DU combined in one system.

6 Conclusions

1. The application of ambiguous spatial-temporal transformations in combination with learning algorithms offers a novel mechanism for encryption and decryption.
2. The combination of unique hardware pairs for encryption- and decryption units with embedded biometric data allows an ultra secure unidirectional communication via public networks.

References

1. O. Baruth, R. Eckmiller and D. Neumann, "Retina encoder tuning and data encryption for learning retina implants," *Proc. of the Int. Joint Conf. on Neural Networks,* (IJCNN) 2003, Vol. **1**, pp. 1249-1252, Portland, Oregon, 2003.
2. E.A. Bernardete and E. Kaplan, *The dynamics of primate M retinal ganglion cells,* Visual Neuroscience, **16** pp. 355-368, 1999.
3. T.C. Clancy, N. Kiyavash and D.J. Lin, "Secure Smartcard-Based Fingerprint Authentication, " *ACM SIGMM 2003 Workshop on Biometrics Methods and Applications,* pp. 45-52, 2003.
4. D.W. Dong, "Spatiotemporal Inseparability of Natural Images and Visual Sensitivities," in *Computational, neural and ecological constraints of visual motion processing,* (page 371-380), Edited by J.M. Zanker and J. Zeil, Berlin: Springer, 2001.

5. J. Daugman, *The importance of being random: statistical principles of iris recognition,* Pattern Recognition, **36** pp. 279-291, 2003.
6. R. Eckmiller, O. Baruth and D. Neumann, "Method and Device for Decryption-Secure Transfer of Data", PCT Patent Application, PCT WO 2004021694.
7. S. Haykin, Editor, *Adaptive Filter Theory,* New Jersey: Prentice Hall, 4th Edition, 2002.
8. R.J. McEliece, Editor, *The Theory of Information and Coding,* Cambridge: Cambridge University Press, 2002.
9. A. Menezes, P. van Oorschot, and S. Vanstone, Editors, *Handbook of Applied Cryptography,* Boca Raton: CRC Press, 1997.
10. T.M. Mitchel, Editor, *Machine Learning,* New York: McGraw Hill, 1997.
11. S. Parbhakar, P. Sharath, and K. Anil, "Biometric Recognition: Security and privacy concerns," *IEEE Security and Privacy Magazine,* **1**, pp. 33-42, 2003.
12. N. Ratha and R. Bolle, Editiors, *Automatic Fingerprint Recognition Systems,* New York: Springer, 2004.
13. B. Schneier, Editor, *Applied Cryptography: Protocols, Algorithms, and Source Code in C,* New York: John Wiley and Sohns, 2nd Edition, 1996.
14. J. Si, A.G. Barto, W.B. Powell, and I.D. Wunsch, Editors, *Handook of Learning and Approximate Dynamic Programming,* Piscataway: IEEE Press and New York: Wiley-Interscience, 2004.
15. C. Soutar. D. Roberge, A. Stoianov, R. Gilroy, and K.V. Kumar, "Biometric Encryption," in *ICSA Guide to Cryptography,* (chapter 22), Edited by R.K. Nichols, McGraw-Hill, 1999.
16. K.R. Rao and P.C. Yip, Editors, *The Transform and Data Compression Handbook,* Boca Raton: CRC Press, 2001.
17. T. Wollinger, J. Guadjardo, and C. Paar, "Security on FPGAs: State-of-the-art implementations and attacks", ACM Transactions in Embedded Computing Systems, Vol. **3**, pp. 534-574, August 2004.
18. T. Wollinger and C. Paar, "Security aspects of FPGAs in cryptographic application" in *New Algorithms, Architectures, and Applications for Reconfigurable Computing,* (chapter 1), Edited by W. Rosenstiel and P. Lysaght, Dordrecht; Boston ;London: Kluwer Academic Publishers, 2004.