

# Optimizing Turbo Codes for Secret Key Generation in Vehicular Ad Hoc Networks

Dhouha Kbaier Ben Ismail<sup>1</sup>, Petros Karadimas<sup>2</sup>, Gregory Epiphaniou<sup>1</sup>, and Haider Al-Khateeb<sup>1</sup>

<sup>1</sup> University of Wolverhampton, Wireless Cyber Research Institute, UK

<sup>2</sup> University of Glasgow, School of Engineering, Glasgow, Scotland

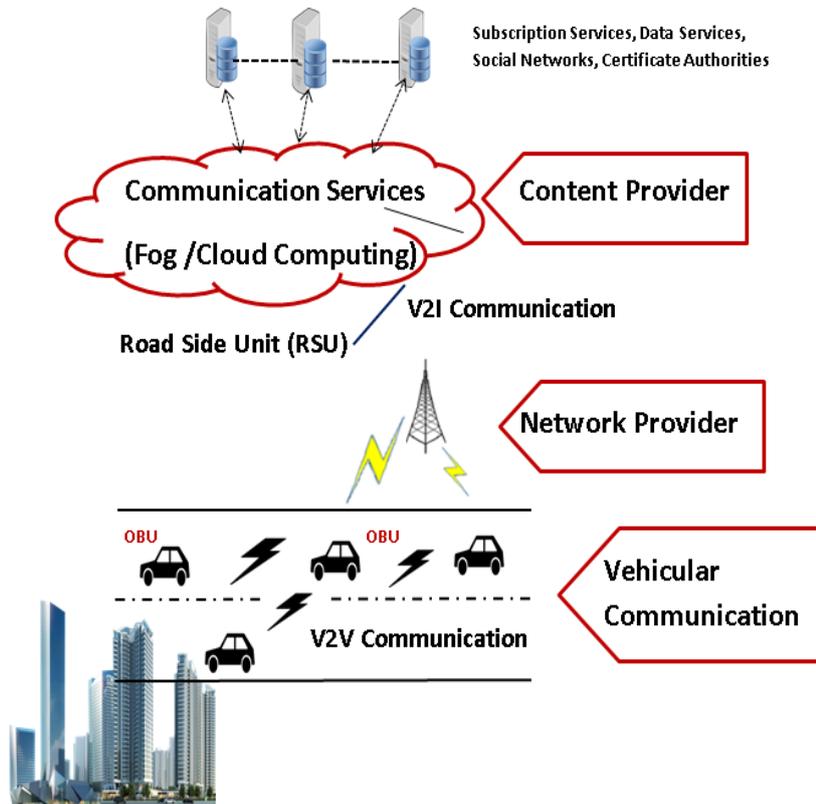
**Abstract.** We present an algorithm that allows two users to establish a symmetric cryptographic key by incorporating the most important features of the wireless channel in vehicle-to-vehicle (V2V) communication. Non-reciprocity compensation is combined with turbo codes (TCs) for error reconciliation purposes. For fair comparisons, the indexing technique is applied in conjunction with the non-reciprocity compensation technique. A series of simulations are run to calculate key performance indicators (KPIs). High entropy values are obtained throughout all rounds of simulation during the key extraction process. Furthermore, simulation results indicate a significant improvement in bit mismatch rate (BMR) and key generation rate (KGR) when TCs are used. Increasing the number of iterations in the TC can significantly improve the Bit Error Rate (BER), thus generating more symmetric keys. The key generation rate was reported high ranging from 17 to 19 for the 256-bit symmetric keys per minute with TCs, while it is ranging from 2 to 5 when compared with a sample indexing technique published in the public domain. Finally, simulations proved also improvements for different key lengths as part of the error reconciliation process when TCs are used with an almost regular permutation (ARP) instead of a random permutation.

**Keywords:** Almost regular permutation, Bit mismatch rate, Entropy, Error reconciliation, Key generation, quantization, Scatterers' mobility, Thresholding, Turbo codes, VANET.

## 1 Introduction

Traditional wireless communications are vulnerable to man-in-the-middle attacks where certain aspects of confidentiality, integrity, and availability are violated. Conventional cryptographic solutions based predominantly on-stream ciphers generate shared secrets using pre-computational techniques or public key cryptography [1]. Public key cryptography, in particular, has proved to increase computational complexity during secret key generation, especially for low-end energy efficient devices [2]. Channel-based key extraction approaches try to exploit the physical properties of wireless channels such as reciprocity and temporal/spatial variability in an attempt to provide the necessary randomness for the symmetric key creation [3], [4].

Vehicular ad hoc network (VANET) based applications are expected to address challenges that current transportation systems are facing, since they can provide solutions for a safer, more efficient and sustainable future intelligent transportation systems (ITS). In a typical VANET environment where access to infrastructure is given (see Fig.1), the wireless links between nodes and co-existent adversaries experience uncorrelated channel attributes. Nodes are also distributed and self-organized with the majority of wireless communication carried out by on-board units (OBU) integrated with additional services and processes running [5]. Therefore, these channels in vehicular networks can offer some level of confidentiality during the key generation process between parties, which reduce the computational complexity and relax certain barriers related to key management requirements.



**Fig. 1.** Vehicular Networking Architecture.

In this paper, all the essential vehicle-to-vehicle (V2V) communication characteristics are incorporated in the key generation process such as three-dimensional (3D) multipath propagation and surrounding scatterers' mobility (i.e. other vehicles). We employ the comprehensive parametric stochastic V2V channel model presented in [6] to synthetically generate the receiver's channel response (Bob's channel), from which the

transmitter's response (Alice's channel) arises after applying the non-reciprocity compensation technique presented in [7]. After the necessary thresholding used to allocate bits according to designated signal levels, we utilize turbo coding (TC) techniques for information reconciliation. We focus on several parameters that affect the performance of TCs such as number of decoding iterations, generator polynomials, constraint lengths of the component encoders and the interleaver type. For fair comparisons, the indexing technique [8] was applied in conjunction with the non-reciprocity compensation technique in [7]. Simulations are run, and performance analysis is carried out. The key generation rate (KGR) and bit mismatch rate (BMR) are computed in both scenarios.

This paper is organized as follows. Section 2 presents an overview of the most important error correction codes that can be potentially used in the information reconciliation stage. Then, the authors present their algorithm and the adopted V2V channel model in section 3. Furthermore, section 4 deals with information reconciliation using TCs in VANET, while section 5 focuses on the importance of privacy amplification. A Graphical User Interface (GUI) was designed to run the developed algorithm. Thus, simulations are run, and performance analysis is carried out in section 6. Several key performance indicators (KPIs) are employed. Finally, section 7 draws some conclusions.

## 2 Error reconciliation

Error reconciliation is the next step in the secret key generation process to correct mismatched information due to imperfect reciprocity and random noise in the channel. The bit mismatch rate is defined as the number of bits that do not match between two devices divided by the total number of bits extracted prior to error reconciliation and privacy amplification. Several error reconciliation algorithms have been introduced with different tradeoffs between communication and computational complexity, throughput error correction capabilities (e.g. Cascade [9] and Winnow [10]). Gallager's Low Density Parity Check (LDPC) codes have recently been shown to increase the rate of error reconciliation with computational overhead added as part of their operation. LDPC can be more efficient than Cascade as they can become rate adaptive leading to more efficient interactive reconciliation protocols [11], [12].

The invention of turbo codes (TCs) [13] was a revival for the channel coding research community. Historical turbo codes, also sometimes called Parallel Concatenated Convolutional Codes (PCCC), are based on a parallel concatenation of two Recursive Systematic Convolutional (RSC) codes separated by an interleaver. They are called "turbo" in reference to the analogy of their decoding principle with the turbo principle of a turbo compressed engine, which reuses the exhaust gas in order to improve efficiency. The turbo decoding principle calls for an iterative algorithm involving two component decoders exchanging information in order to improve the error correction performance with the decoding iterations. This iterative decoding principle was soon applied to other concatenations of codes separated by interleavers, such as Serial Concatenated Convolutional Codes (SCCCs) [14], [15], sometimes called serial turbo codes, or concatenation of block codes, also named block turbo codes [16], [17]. The near-

capacity performance of turbo codes and their suitability for practical implementation explain their adoption in various communication standards. In [18] the authors proposed utilizing Turbo codes for reconciliation purposes. Further investigation in [19] show that TCs are good candidates for reconciliation. The efficacy of TCs with regards to their error correction capabilities in various wireless communication standards is also recorded in [20]. Further work in [21] demonstrates the improved performance of TCs over Reed Solomon and CCs which are the de-facto error correction codes used in 802.11p vehicular networks. However, this work does not comprehensively incorporate physical propagation characteristics such as 3D scattering and scatterers' mobility which is addressed in this work.

### 3 Detailed architectural design

Fig. 2 presents the algorithm's strawman. Starting from the first step, synthetic data are generated for the purpose of demonstration by employing the Monte Carlo simulation method [22], [23]. The input parameters of the algorithm are provided by the inherent physical attributes of the dynamic V2V propagation channel. Indeed, the comprehensive parametric stochastic V2V channel model presented in [6] is adopted. The vehicles are in motion and all are generally equipped with both Tx and Rx.

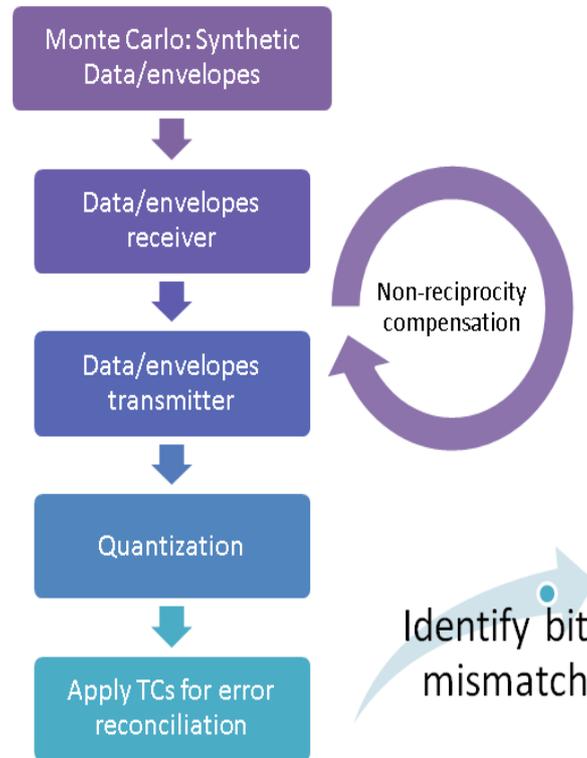


Fig. 2. Detailed Architectural Design.

Based on that channel model, the receiver's samples (Bob's channel estimates) are generated. From the received data, the transmitted data (Alice's channel estimates) are modeled by considering the non-reciprocity compensation technique presented in [7]. At this phase, a lossy quantization process is preferred due to its computational simplicity. The target is to end up with a maximum secret bit extraction rate and entropy. The number of bits that do not match between Alice and Bob to the number of bits extracted by the quantization process determines the bit mismatch rate. Turbo decoding is then performed in order to generate a symmetric output, i.e. symmetric keys for Alice and Bob. Performance of the reconciliation method can be evaluated by measuring the BMR and the Bit Error Rate (BER). For fair comparisons, the indexing technique was applied in conjunction with the non-reciprocity compensation technique.

#### **4 Information reconciliation using Turbo Codes in VANET**

The invention of turbo codes (TCs) [13] was a revival for the channel coding research community. The near-capacity performance of turbo codes and their suitability for practical implementation explain their adoption in various communication standards as early as the late nineties: firstly, they were chosen in the telemetry coding standard by the CCSDS (Consultative Committee for Space Data Systems) [24], and for the medium to high data rate transmissions in the third generation mobile communication 3GPP (Third Generation Partnership Project)/UMTS (Universal Mobile Telecommunications System) standard [25]. They have further been adopted as part of the Digital Video Broadcast - Return Channel via Satellite and Terrestrial (DVB-RCS and DVB-RCT) links [26, 27], thus enabling broadband interactive satellite and terrestrial services. More recently, they were also selected for the next generation of 3GPP2/cdma2000 wireless communication systems [28] as well as for the IEEE 802.16 standard (WiMAX) [29] intended for broadband connections over long distances. In our approach, we use TCs for reconciliation purposes in VANET environment. In section 6, simulations are run, and performance analysis is carried out.

#### **5 Privacy Amplification**

Masquerading or eliminating information exchanged during this process is usually defined as privacy amplification [30, 31]. Privacy amplification is used to transform a string which is only partially secret to a highly secret string usually shorter. This process is also used to account for any information exposed during error reconciliation phase and ensure that eavesdroppers do not gain significant advantage to the point where they are able to reconstruct a significant part of the key. The Winnow protocol discards certain bits during error reconciliation (privacy maintenance) whereas Cascade, LDPC, and TCs do not. In Cascade, the binary search must rerun on previous blocks, whereas for LDPC and TCs only one pass is necessary so there is no need to discard bits before the error reconciliation is complete. The number of bits is tracked and then subtracted from the final key reconciled. The last step in the key generation process assumes that the information extraction about the shared key used should be computationally

expensive to adversaries. Most of the existing approaches focus on different threat models and assumptions around the level of access to the channel. ‘Trapdoor’ functions are used as means to assure a certain level of authentication and integrity in this process. These functions are also used as a mean to reduce the size of the final key and amplify any errors if hashing a reasonable copy of the key is attempted, to a degree that even exhaustive search of the key space would be infeasible.

## 6 Simulations results

The authors have designed a Graphical User Interface (GUI) to run the developed algorithm. Fig. 3 illustrates the main KeyGen simulation with a visualization of the whole process towards the establishment of symmetric keys. It also illustrates samples for both Bob and Alice based on the parametric stochastic model introduced in this work generating the synthetic data for different scenarios in V2V communications. The algorithm currently avoids using statistical measures and it is solely based on the channel fading process. The parameters are fully customizable from the tools menu of the software demo. The GUI is not fundamental for the core algorithmic operation and can be omitted in real-life implementations or fabricated products. The thresholding scheme employed influences the number of samples discarded from both Alice and Bob. The GUI output also demonstrates the number of keys established during the simulation and the total time required generating those keys.

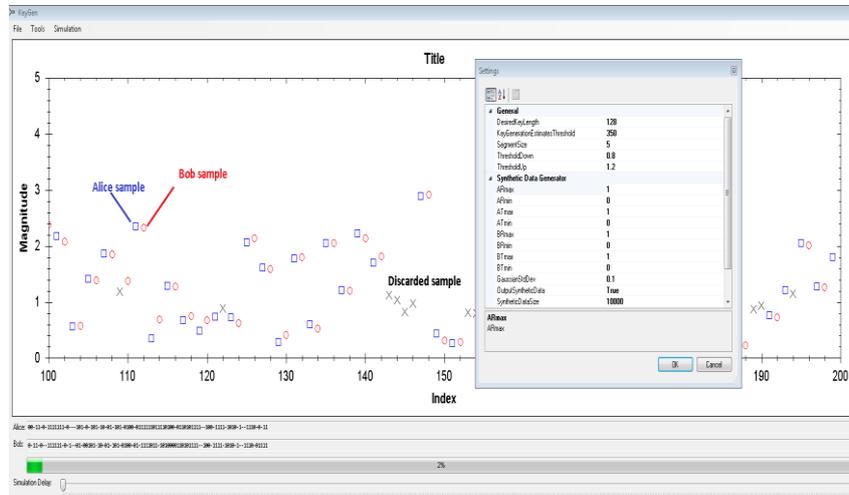


Fig. 3. An implementation of Alice’s and Bob’s samples.

### 6.1 Key Performance Indicators

First, the probing rate for both Alice and Bob  $f_P = f_{PA} = f_{PB}$  are considered the same for the purpose of channel estimates collection. The core Key Performance

Indicators (KPIs) of interest in our protocol up to this stage are the key generation rate, the randomness of the generated bits for symmetric keys and bit mismatch rate (BMR). The entropy is the de-facto metric which quantifies the uncertainty of the generated bit string. The higher the entropy, the limited the ability to deduce a secret key established by Eve is. Usually, BMR is measured as a ratio of the number of bits that do not match between Alice and Bob to the number of bits extracted at the thresholding stage.

## 6.2 Our methodology

Bob's generated sequence after quantization is fed to the input of a TC. During this process, a single threshold is adopted as a lossless quantization scheme with the potential to substantial increase the key generation rate [32]. Turbo decoding is then performed in order to generate a symmetric output, i.e. symmetric keys for Alice and Bob. Performance of the reconciliation method can be evaluated by measuring the BMR and the Bit Error Rate (BER). The comparison is made against the sample indexing technique already applied in our algorithm as discussed in section 1. We calculated BMR by considering the discarded indexes after Alice's and Bob's channel probing. Thus, the BMR is measured as a ratio of the number of bits that do not match between Alice and Bob to the number of bits extracted by the adopted quantization process after appropriate thresholding.

## 6.3 Simulation results

In Table 1, we compute also the key generation rate for different key lengths. TCs are first run with a random permutation and one decoding iteration. Compared to the samples' indexing method in [8], there was a significant improvement on both BMR and key generation rate. The BMR with single thresholding is only 0.02 whereas the estimated BMR with the indexing technique is around 0.22. Note that the BMR with the indexing technique is nearly the same for different key lengths which is coherent with the uniform method used by authors and algorithm presented in [8]. The key generation rate was also reported high considering different key lengths requested. For instance, the secret key rate to generate the 256-bit symmetric key is 17 good keys per minute with TCs while it varies from 2 to 5 symmetric keys per minute with the indexing technique.

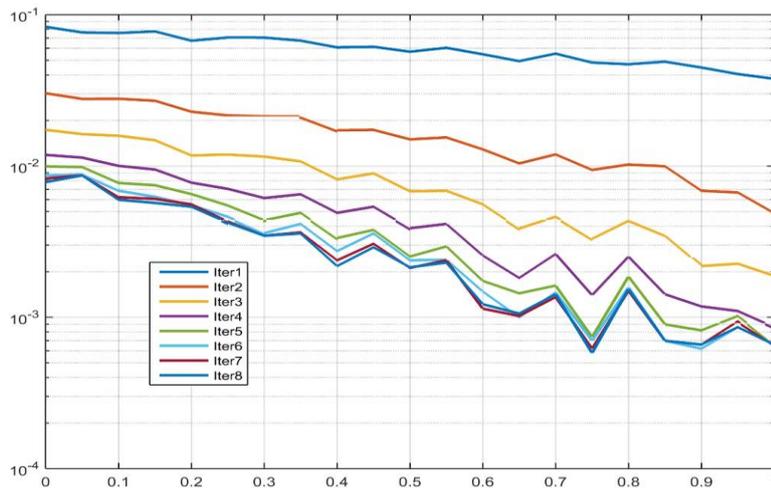
**Table 1.** Simulation results in secret key generation.

|                       | Indexing technique |        |        | Turbo Codes  |     |     |
|-----------------------|--------------------|--------|--------|--------------|-----|-----|
|                       | 128                | 256    | 512    | 128          | 256 | 512 |
| Key length (bits)     | 128                | 256    | 512    | 128          | 256 | 512 |
| BMR                   | 0,22               |        |        | 0,02         |     |     |
| Entropy (bits/sample) | 0,85 to 0,97       |        |        | 0,94 to 0,99 |     |     |
| KGR (keys/min)        | 3 to 7             | 2 to 5 | 1 to 2 | 35           | 17  | 8   |

Then, in Table 2 we compute the key generation rate where TCs are run with an almost regular permutation (ARP) permutation and several decoding iterations. Indeed, increasing the number of iterations in the TC can significantly improve the BER, thus generating more symmetric keys. However, a compromise should be found since this operation is computationally expensive and adds a delay in the process. We report the BER performance of the TC for the block size 5000 bits, at coding rate in Fig. 4. Thus, the total number of samples is 10000 bits for both Alice and Bob. For a signal to noise ratio SNR=0.8 dB, the simulated BER to generate a symmetric shared key between Alice and Bob after error reconciliation is estimated to  $4,9 \times 10^{-2}$  for the 1<sup>st</sup> iteration using TCs while the BER is  $2,5 \times 10^{-3}$  for the 4<sup>th</sup> iteration and only  $1,8 \times 10^{-3}$  for the 8<sup>th</sup> iteration. Note that the KGR reaches a limit and remains the same beyond 4 decoding iterations. In the simulations of Table 2, the maximum number of turbo decoding iterations is set to 4 iterations. As shown in Table 2, simulations proved also improvements for different key lengths as part of the error reconciliation process when TCs are used with an ARP permutation instead of a random permutation. Finally, high entropy values were obtained throughout all rounds of simulation during the key extraction process. Note that the higher the entropy, the limited the ability to deduce a secret key established by an adversary such as Eve.

**Table 2.** Key generation rate with ARP permutation and different decoding iterations.

| Key length (bits)  | Turbo Codes |     |     |
|--------------------|-------------|-----|-----|
|                    | 128         | 256 | 512 |
| Iteration 1        | 36          | 18  | 9   |
| Iteration $\geq 4$ | 39          | 19  | 9   |



**Fig. 4.** BER performance of TC for a block length of 5000 bits. All simulations use the BCJR algorithm with 8 decoding iterations.

In future studies, we would like to further investigate TCs for error conciliation purposes. We will focus on other parameters that affect the performance of TCs such as component decoding algorithms, generator polynomials and constraint lengths of the component encoders.

## 7 Conclusion

We proposed an algorithm considering the most important features in V2V communication such as 3D multipath propagation and surrounding scatterers' mobility. Synthetic data were generated for the purpose of a demonstration by employing the Monte Carlo simulation method. Simulations were run successfully by combining non-reciprocity compensation with turbo codes. Increasing the number of iterations in the TC can significantly improve the BER, thus generating more symmetric keys. Moreover, simulations proved improvements for different key lengths as part of the error reconciliation process when TCs are used with an ARP permutation instead of a random permutation. Compared with a sample indexing technique in the public domain, results have shown significant improvements for key generation rate and bit mismatch rate with high entropy values obtained throughout all rounds of simulation.

## References

1. M. J. B. Robshaw and O. Billet, Eds., *New Stream Cipher Designs - TheeSTREAM Finalists*, ser. Lecture Notes in Computer Science. Springer, 2008, vol. 4986.
2. N. K. Jha, A. Raghunathan, N. R. Potlapally, and S. Ravi, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," *IEEE Transactions on Mobile Computing*, vol. 5, no. undefined, pp. 128–143, 2006.
3. A. Mukherjee, S. A. A. Fakoorian, J. Huang, and A. L. Swindlehurst, "Principles of Physical layer security in multiuser wireless networks: A survey," *CoRR*, vol. abs/1011.3754, 2010. [Online]. Available: <http://arxiv.org/abs/1011.3754>
4. Y. E. H. Shehadeh and D. Hogrefe, "A survey on secret key generation mechanisms on the physical layer in wireless networks," *Sec. and Commun. Netw.*, vol. 8, no. 2, pp. 332–341, Jan. 2015. [Online]. Available: <http://dx.doi.org/10.1002/sec.973>
5. F. Qu, Z. Wu, F. Y. Wang, and W. Cho, "A security and privacy review of vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2985–2996, Dec 2015.
6. P. Karadimas and D. W. Matolak, "Generic stochastic modeling of vehicle-to-vehicle wireless channels," *Vehicular Communications*, vol. 1, no. 4, pp. 153–167, 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.vehcom.2014.08.001>
7. H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response." in *INFOCOM*. IEEE, 2013, pp. 3048–3056.
8. S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, *Secret Key Extraction from Level Crossings over Unauthenticated Wireless Channels*. Springer US, 2010, pp. 201–230. [Online]. Available: [http://dx.doi.org/10.1007/978-1-4419-1385-2\\_9](http://dx.doi.org/10.1007/978-1-4419-1385-2_9)
9. G. Brassard and L. Salvail, "Secret-Key Reconciliation by Public Discussion," in *Eurocrypt '93*. Springer-Verlag, 1993, pp. 410–423. [Online]. Available: <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.9686>

10. W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, and C. G. Peterson, "Fast, efficient error reconciliation for quantum cryptography," *Phys. Rev. A*, vol. 67, p. 052303, May 2003. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevA.67.052303>
11. J. Mart'inez-Mateo, D. Elkouss, and V. Martin, "Blind reconciliation," *Quantum Information & Computation*, vol. 12, no. 9-10, pp. 791–812, 2012. [Online]. Available: <http://www.rintonpress.com/xxqic12/qic-12-910/0791-0812.pdf>
12. J. Martnez-Mateo, D. Elkouss, and V. Martn, "Interactive reconciliation with low-density parity-check codes," in *2010 6th International Symposium on Turbo Codes Iterative Information Processing*, Sept 2010, pp. 270–274.
13. C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC'93, Geneva, Switzerland*, vol. 2, May 1993, pp. 1064–1070.
14. S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 909–926, May 1998.
15. S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electronics letters*, vol. 32, no. 13, pp. 1186–1188, June 1996.
16. —, "Serial concatenation of block and convolutional codes," *Electronics Letters*, vol. 32, no. 10, pp. 887–888, May 1996.
17. R. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Transactions on Communications*, vol. 46, no. 8, pp. 1003–1010, 1998.
18. K. Nguyen, G. V. Assche, and N. J. Cerf, "Side-information coding with turbo codes and its application to quantum key distribution," *CoRR*, vol. cs.IT/0406001, 2004. [Online]. Available: <http://arxiv.org/abs/cs.IT/0406001>
19. N. Benletaief, H. Rezig, and A. Bouallegue, "Toward efficient quantum key distribution reconciliation," *Journal of Quantum Information Science*, vol. 2014, 2014.
20. E. Yeo and V. Anantharam, "Iterative decoder architectures," *IEEE Communications Magazine*, vol. 41, no. 8, pp. 132–140, Aug 2003.
21. G. Kiokes, G. Economakos, A. Amditis, and N. K. Uzunoglu, "A comparative study of ieee 802.11 p physical layer coding schemes and fpga implementation for inter vehicle communications," *Modern Traffic and Transportation Engineering Research*, vol. 2, no. 2, pp. 95–102, 2013.
22. P. Hirschhausen, L. Davis, D. Haley and k. Lever, "Identify key design parameters for Monte Carlo simulation of Doppler Spread channels," in *Communications Theory Workshop (AusCTW)*, Sydney, 2014.
23. P. Hoecher, "A statistical discrete-time model for the WSSUS multipath channel," *IEEE Transactions on vehicular technology*, vol. 41, no. 4, 1992.
24. T. Synchronization and C. Coding. Recommendation for Space Data System Standards. Technical report, CCSDS 131.0-B-1. Blue Book.
25. Third Generation Partnership Project (3GPP) Technical SpecicationGroup, Multiplexing and channel coding (FDD), June 1999, TS 25.212, v2.0.0.
26. DVB, Interaction channel for satellite distribution systems, 2000, ETSI EN 301 790, v. 1.2.2.
27. DVB, Interaction channel for digital terrestrial television, 2001, ETSI EN 301 958, v. 1.1.1.
28. Third Generation Partnership Project 2 (3GPP2), Physical layer standard for cdma2000 spread spectrum systems, Release D, February 2004, 3GPP2 C.S0002-D, Version 1.0.
29. IEEE standard for local and metropolitan area networks. Part 16: air interface for xed broadband wireless access systems, November 2004, IEEE 802.16-2004.

30. Y. E. H. Shehadeh and D. Hogrefe, "A survey on secret key generation mechanisms on the physical layer in wireless networks," *Security and Communication Networks*, vol. 8, no. 2, pp. 332–341, 2015.
31. T. Wang, Y. Liu, and A. V. Vasilakos, "Survey on channel reciprocity based key establishment techniques for wireless systems," *Wireless Networks*, vol. 21, no. 6, pp. 1835–1846, 2015.
32. B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 401–410. [Online]. Available: <http://doi.acm.org/10.1145/1315245.1315295>