Secure Physical-layer Key Generation Protocol and Key Encoding in Wireless Communications

Apirath Limmanee and Werner Henkel
Jacobs University
EECS, TrSys, Campus Ring 1
28759 Bremen, Germany
Email: {a.limmanee, w.henkel}@jacobs-university.de

***Abstract*—Wireless communication between two nodes in a multi-path, multi-hop network does not require that the forward transmission path and the reverse one are the same. However, if a physical-layer security scheme based on mutual channel-state-information (CSI) [3],[4] is used, they at least have to make sure that they use the same channels for secret-key generation. A protocol is proposed in this paper to achieve that purpose in a secure manner.**

An information-theoretic analysis of physical-layer key generation given in [2] states that there are vulnerable key symbols that might be estimated by eavesdroppers. In this paper, we show how to protect those key symbols by means of physical-layer key encoding. We provide necessary and sufficient conditions on the code in order to achieve perfect secrecy, even when the eavesdropper knows the code, and derive the asymptotic code rate accordingly. We also discuss how to design the code when the number of vulnerable key symbols are not known.

I. Introduction

Massey [1] suggests a model of a secret-key cryptosystem as shown in Fig. 1. Its concept of security is based on Shannon's idea of perfect secrecy, which means that, for a plain text $\mathbf{X}$ and its cryptogram $\mathbf{Y}$, $P(\mathbf{X}=\mathbf{x}|\mathbf{Y}=\mathbf{y}) = P(\mathbf{X}=\mathbf{x})$ for all possible plain texts $\mathbf{x} = [x_1, x_2, ..., x_M]$ and cryptograms $\mathbf{y}$. In this case, neither the knowledge of the cryptogram $\mathbf{y}$ nor large computational power can help an enemy cryptanalyst to decrypt the message $\mathbf{x}$, unless he or she knows the secret key.

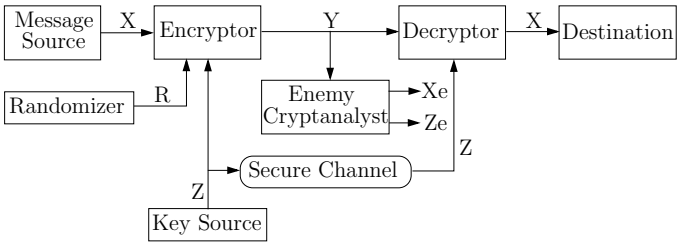

Fig. 1. A secret-key cryptosystem

In Massey's generalized model, the encryptor mixes the plain text $\mathbf{X}$ with the random message $\mathbf{R}$ and the secret key $\mathbf{Z}$ to achieve perfect secrecy. However, some systems, such as the "one-time pad," which will be discussed later, do not require $\mathbf{R}$. The major difficulty in the implementation of such cryptosystems lies in the secure distribution of the key. There is recently a new idea of transforming wireless channel state information (CSI) into secret keys. It is widely known that wireless channel coefficients, characterized by their phases and amplitudes, depend heavily on the location, the environment, and the movement of the transmitter and the receiver to the extent that other terminals except the two can predict almost nothing about their channel parameters, and hence their secret key. The generation of the secret key consists of two steps, deriving the channel estimates before quantizing them into secure key symbols. We refer to this technique as "wireless physical-layer secret-key generation." This must be distingished from the term "physical layer security" used in [5], which requires that the legitimate users' channels have SNR advantage over those of eavesdroppers.

Channel estimates can be derived using a known pilot sequence, which is transmitted back and forth between the transmitter and the receiver such that they can learn about channel coefficients from the symbols distorted by the channel. The outcome of the channel estimation process is a set of complex channel coefficients which must be quantized into secret key symbols, as shown in Fig. 2. The process of key generation is discussed in detail in [2],[3].

Since, in a generalized wireless network, there can be several channels that the pilot sequence can take to travel from one node to another, the technique implicitly assumes a bi-lateral agreement between the two nodes regarding the channel to be used for key generation. Earlier works did not tell how to form such an agreement in a secure manner, we will do it in the next section.

Although the enemy cryptanalyst is not located at the same place as the receiver during transmission, he or she may have been there before and it is possible that channel coefficients (especially the amplitudes) do not change much. Therefore, it is wise to assume that he or she can correctly predict some key symbols and we should try to find some countermeasures. To do so, we use a similar concept to Shamir's secret sharing [7] and Cai and Yeung's secure network coding [8]. We call our scheme "physical-layer key encoding," which is designed such that, up to a certain threshold, partial knowledge of key symbols derived from the channel leaves the encoded key completely undetermined. This is performed at the transmitter prior to the one-time pad encryptor block in our wireless physical-layer secret-key cryptosystem shown in Fig. 2 and will be discussed in detail in Section III.

978-1-4244-8865-0/10/$26.00 ©2010 IEEE

In Section IV, we suggest how to set the number of vulnerable bits based on an equivalent model of the eavesdropper. Section V then concludes the paper.

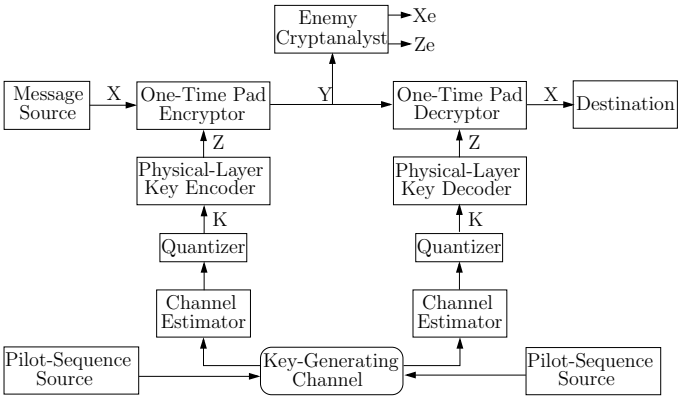

Fig. 2. A modified secret-key cryptosystem based on mutual CSI with physical-key encoding and decoding

II. Wireless Physical-Layer Secret-Key Generation Protocol

In this section, we propose a simple scheme to guarantee that the receiver derives the secret key from the same channel as the transmitter does. The advantage of this scheme is that it supports datagram networks in which the transmission route might not be known in advance by the transmitter and the receiver. The transmitter and the receiver always generate the secret key from the same channel without knowing from which channel they do. This lack of explicit routing information makes life very hard for the eavesdropper.

The scheme proceeds as follows. First, the transmitter passes a complete packet containing a pilot sequence to a next node on the way to the receiver and broadcasts a packet header to all other neighbors. That next node follows the same procedure regarding its following node and other neighbors. This repeats until the packet reaches the receiver, who uses it to estimate the channel parameters and derive the secret key. After that, a packet with the same pilot sequence is sent back along the same path. To do so, the receiver sends it to every neighbor, who, after reading the header, only sends the packet further upstream if it belongs to the forward path. Again, this repeats until the transmitter gets the packet back and derives the secret key accordingly.

To illustrate the scheme, let us consider the butterfly network in Fig. 3. If we consider $ACFGD$ as a forward path from A to D, the scheme proceeds as follows.

Forward:
1. A sends a packet to C and its header also to B.
2. C sends the packet to F and its header also to E.
3. F sends the packet to G and its header to B.
4. G sends the packet to D and its header also to E.

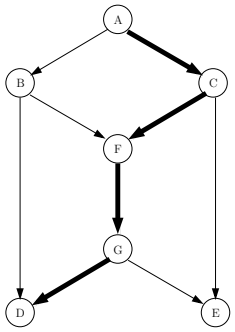

Fig. 3. A butterfly network

Reverse:
1. D sends the packet to B and G.
2. After reading the header, B discards the packet, whereas G sends it to F and E.
3. After reading the header, E discards the packet, whereas F sends it to B and C.
4. After reading the header, B discards the packet, whereas C sends it to A and E.
5. After reading the header, E discards the packet, whereas A successfully receives it, together with the correct channel information.

III. Physical-Layer Key Encoding for One-Time-Pad Encryptor

In this section, we shall present physical-layer key encoding which generates "the encoded key" $\mathbf{Z} = [Z_1, Z_2, ..., Z_J]$ as a codeword from "an original quantized key" $\mathbf{K} = [K_1, K_2, ..., K_{I_K}]$ and feeds it into the one-time pad encryptor. The encoder aims at protecting the secret key in case the enemy can correctly estimate some channel coefficients, and hence some original quantized key symbols. Three parameters, I_K, I_{SK}, and I_{VK} are of interest to the encoder. The first one, I_K, is the number of symbols that can be generated by the quantizer in Fig. 2. The second one, I_{SK}, is the number of secure key symbols that the enemy cannot correctly estimate, whereas the third, I_{VK}, is the number of vulnerable ones that he or she can correctly estimate, such that $I_K = I_{SK} + I_{VK}$.

The information-theoretic derivation of I_K, I_{SK}, and I_{VK} is discussed in [2] but not in this paper, where we are more interested in the following questions: Given I_K, I_{SK}, and I_{VK}, what is the important property of the physical-layer key encoding that ensures perfect secrecy as well as efficiency? How can we choose the code rate? And how can we derive the optimal code? Some questions will be answered by theorems 1-3 proposed in this section. Before going that far, we will first describe the one-time pad encryptor.

Consider a non-randomized cipher in which elements in the plaintext $\mathbf{X} = [X_1, X_2, ..., X_M]$, ciphertext $\mathbf{Y} = [Y_1, Y_2, ..., Y_N]$, and secret key $\mathbf{Z} = [Z_1, Z_2, ..., Z_J]$ all take values in the L-ary alphabet and $J = N = M$. Suppose that the key is chosen to be completely random, i.e., $P(Z_i = z) =$

L^{-M}, $i = 1, 2, ..., M$, for all possible values z of the secret key, and that the enciphering transformation is

$$Y_i = (X_i + Z_i) \bmod M,\ i = 1, 2, ..., M. \tag{1}$$

Since, for each possible choice x and y of X_i and Y_i, respectively, there is a unique z such that $Z_i = z$ satisfies (1), it follows that $P(Y_i = y|X_i = x) = L^{-M}$ for every particular y and x, no matter what the statistics of X_i may be. Thus, X_i and Y_i are statistically independent, and hence this system provides perfect secrecy [1]. The system is called a modulo-L Vernam system or one-time pad and is used in our model in Fig. 2 to combine the message and the encoded key together.

Since, out of the total I_K quantized key symbols, I_{VK} symbols are vulnerable symbols, we can see that, had a one-time pad encryptor been used without physical-key encoding, I_{VK} ciphertext symbols would have been decrypted by the eavesdropper. Thus, in order to construct a secure Y_1, we use the following linear combination for Z_1.

$$Z_1 = (K_1 + K_2 + ... + K_{I_{VK}+1}) \bmod M \tag{2}$$

After substituting (2) into (1) using $i = 1$, we can see that even if the set of all vulnerable key symbols is a subset of $\{K_1, K_2, ..., K_{I_{VK}+1}\}$, there is still one symbol that is unknown to the eavesdropper. If every key symbol is statistically independent of others, perfect secrecy of Y_1 is achieved.

Now, in order to construct Z_2, Z_3, and so on, we perform a linear combination of I_{VK} key symbols similar to (2). Therefore, our physical-layer key encoding can be defined by a linear block code C_p transforming an I_K-tuple $\mathbf{K}$ over $GF(q^{I_K})$ of key symbols obtained from the quantizer into an M-tuple codeword $\mathbf{Z}$ over $GF(q^M)$. We propose the following theorem providing two necessary and sufficient conditions for perfect secrecy.

Theorem 1: If I_{VK} out of I_K physical-layer key symbols generated by the quantizer can be correctly estimated by the eavesdropper, the cryptosystem in Fig. 2 still maintains perfect secrecy if and only if each member in the I_K-dimensional vector $\mathbf{K}$ is independent of one another and the physical-layer code C_p has the following properties.
1.1. Every codeword has a Hamming weight of at least $I_{VK} + 1$.
1.2. Every linear combination of any subset of codewords gives a Hamming weight of at least $I_{VK} + 1$.

Proof From earlier discussion it should be clear that the first condition is necessary. The necessity of the second condition can be proved by contradiction as follows. If the combination of ν codewords $\sum_{i=1}^{\nu} Z_i$ gives a Hamming weight of $I_{VK} + 1 - \delta$, where δ is a positive integer, the eavesdropper who calculates $\sum_{i=1}^{\nu} Y_i$ may know the exact value of $\sum_{i=1}^{\nu} Z_i$ since the Hamming weight of the combination does not exceed the number of vulnerable symbols. If $\sum_{i=1}^{\nu} Z_i$ is known, perfect secrecy is not achieved because these ν encoded key symbols are not independent.

As for the sufficiency, we can also prove it by contradiction. Now, we have to prove that if perfect secrecy is not achieved, either the condition 1.1 or 1.2 is not satisfied. By definition, perfect secrecy in a one-time-pad system is not achieved only if either 1) at least one Z_i, $i = 1, 2, ..., M$ is known or 2) any linear combination of some Z_i is known implying linear dependence among key symbols. Since 1) and 2) will not happen if the condition 1.1 and 1.2 are, respectively, satisfied, the two conditions are sufficient.

Now, we focus our attention on the special case when $\mathbf{K}$ and $\mathbf{Z}$ are vectors of binary data and present the second theorem.

Theorem 2: If $\mathbf{K} \in GF(2^{I_K})$, in order to generate $\mathbf{Z} \in GF(2^n)$, which is a codeword of n encoded key bits providing perfect secrecy in our system, the following conditions on I_K are necessary and sufficient.
2.1. If $I_{VK} + 1$ is even,

$$I_K \geq \frac{(n+1)}{2}(I_{VK} + 1) \tag{3}$$

2.2 If $I_{VK} + 1$ is odd,

$$I_K \geq \frac{(n+1)}{2}(I_{VK} + 1) + \frac{(n-1)}{2} \tag{4}$$

Proof We prove this theorem by mathematical induction. We first demonstrate that the theorem is valid for $n = 1$ and $n = 2$ before showing that if the theorem is valid for any n, it will hold true for $n + 1$. The case of $n = 1$ can be easily validated by substituting it into (3) and (4) and observing that the resulting I_K corresponds to that in Theorem 1.

Given Z_1 in Eq. (2), when $I_{VK} + 1$ is an even number. Let

$$Z_2 = K_{\frac{1}{2}(I_{VK}+3)} \oplus K_{\frac{1}{2}(I_{VK}+5)} \oplus ... \oplus K_{\frac{3}{2}(I_{VK}+1)}. \tag{5}$$

Now, the generator matrix generating Z_1 and Z_2 can be written as follows.

$$\mathbf{G_p} = \begin{bmatrix} \overbrace{1\ 1\ ...\ 1\ 1\ 1\ ...\ 1}^{I_{VK}+1} & \overbrace{0\ 0\ ...\ 0\ 0\ 0\ ...\ 0}^{I_K - I_{VK} - 1} \\ \underbrace{0\ 0\ ...\ 0}_{\frac{1}{2}(I_{VK}+1)} & \underbrace{1\ 1\ ...\ 1\ 1\ 1\ ...\ 1}_{I_{VK}+1}\ 0\ 0\ ...\ 0 \end{bmatrix}^T = [\mathbf{g}_1\ \mathbf{g}_2] \tag{6}$$

such that

$$\mathbf{Z} = [Z_1, Z_2] = \mathbf{K} \cdot \mathbf{G_p}, \tag{7}$$

where

$$\mathbf{K} = [K_1, K_2, ..., K_{I_K}]. \tag{8}$$

When $n = 2$, i.e., only two encoded key bits are to be constructed, we can see from (6) that $I_K = \frac{3}{2}(I_{VK} + 1)$ original key bits from the quantizer are sufficient, i.e., the zero padding after the $\frac{3}{2}(I_{VK} + 1)^{th}$ is trivial. This sufficient value of I_K holds for any given $\mathbf{g}_1$ with Hamming weight of $I_{VK} + 1$ because, in order to choose $I_{VK} + 1$ rows of $\mathbf{g}_2$ with 1-valued elements, we should have $\frac{I_{VK}+1}{2}$ rows where those

elements of $\mathbf{g}_1$ in the same positions have the value 1. In other words, if we pile $\mathbf{g}_2$ up onto $\mathbf{g}_1$, at most $\frac{I_{VK}+1}{2}$ positions of 1-valued elements may overlap. If there are more overlapping positions, the second condition in Theorem 1 will be violated. If there are less overlapping positions, no condition is violated, but it is not an economical use of original quantized key bits because I_K must now be greater than $\frac{3}{2}(I_{VK}+1)$.

We can therefore conclude that with $n = 2$ and an even $I_{VK}+1$, $I_K \geq \frac{(n+1)}{2}(I_{VK}+1)$ is a necessary and sufficient condition on I_K. If we follow the same line of reasoning with $n > 2$, we see that when each $\mathbf{g}_{i+1}$ is piled up onto the heap of $\mathbf{g}_j$, $j = 1, 2, ..., i$, the most economical way in terms of the number of original quantized key bits used is still to have $\frac{I_{VK}+1}{2}$ 1-valued elements in overlapped positions with any 1-valued elements in any of those $\mathbf{g}_j$, $j = 1, 2, ..., i$. This means at least $\frac{I_{VK}+1}{2}$ more original quantized key bits are needed for each increment of n, thus completing the proof of 2.1.

A similar line of reasoning can be used to prove 2.2. With odd $I_K + 1$, $\mathbf{G_p}$ in (6) becomes

$$\mathbf{G_p} = \begin{bmatrix} \overbrace{1\ 1\ ...\ 1\ 1\ ...\ 1}^{I_{VK}+1} & \overbrace{0\ 0\ ...\ 0\ 0\ 0\ ...\ 0}^{I_K-I_{VK}-1} \\ \underbrace{0\ 0\ ...\ 0}_{\frac{1}{2}(I_{VK}+2)} & \underbrace{1\ 1\ ...\ 1\ 1\ 1\ ...\ 1}_{I_{VK}+1}\ 0\ 0\ ...\ 0 \end{bmatrix}^T = [\mathbf{g}_1\ \mathbf{g}_2] \tag{9}$$

With the overlapped positions reduced from $\frac{I_{VK}+1}{2}$ in the even case to $\frac{I_{VK}}{2}$ in the odd case, the value of necessary I_K for each n increases. One can verify Eq. (4) when $n = 2$ by looking at the $\mathbf{G_p}$ in (9). With $n \geq 2$, at least $\frac{I_{VK}+2}{2}$ more original quantized key bits are needed for each increment of n, thus completing the proof of 2.2.

The next theorem concerns the asymptotic rate of the code.

Theorem 3: The minimum asymptotic code rate of C_p as $n \to \infty$ is $\frac{I_{VK}+1}{2}$ if $I_{VK}+1$ is even. Otherwise, it is $\frac{I_{VK}+2}{2}$.

Proof The code rate is the ratio between the number of originial bits to that of encoded ones. Therefore, the asymptotic code rate as $n \to \infty$ is derived by dividing (3) and (4) by n and finding the limit of the results as as $n \to \infty$.

From the proof of Theorem 2, we have derived our generator matrix prototype of a physical-layer key encoder for any value of I_{VK}, when $I_{VK}+1$ is even or odd in equations (10) or (11), respectively.

$$\mathbf{G_p} = \begin{bmatrix} \overbrace{1\ 1\ ...\ 1\ 1\ ...\ 1}^{I_{VK}+1} & \overbrace{0\ ...\ 0\ 0\ ...\ 0\ 0\ 0\ ...\ 0}^{I_K-I_{VK}-1} \\ \underbrace{0\ 0\ ...\ 0}_{\frac{1}{2}(I_{VK}+1)} & \underbrace{1\ ...\ 1\ 1\ ...\ 1}_{I_{VK}+1}\ 0\ ...\ 0\ 0\ 0\ ...\ 0 \\ & \vdots \\ \underbrace{0\ 0\ ...\ 0\ 0\ ...\ 0\ 0\ ...\ 0\ 0\ ...\ 0}_{I_K-I_{VK}-1} & \underbrace{1\ 1\ ...\ 1}_{I_{VK}+1} \end{bmatrix}^T \tag{10}$$

$$\mathbf{G_p} = \begin{bmatrix} \overbrace{1\ 1\ ...\ 1\ 1\ ...\ 1}^{I_{VK}+1} & \overbrace{0\ ...\ 0\ 0\ ...\ 0\ 0\ 0\ ...\ 0}^{I_K-I_{VK}-1} \\ \underbrace{0\ 0\ ...\ 0}_{\frac{1}{2}(I_{VK}+2)} & \underbrace{1\ ...\ 1\ 1\ ...\ 1}_{I_{VK}+1}\ 0\ ...\ 0\ 0\ 0\ ...\ 0 \\ & \vdots \\ \underbrace{0\ 0\ ...\ 0\ 0\ ...\ 0\ 0\ ...\ 0\ 0\ ...\ 0}_{I_K-I_{VK}-1} & \underbrace{1\ 1\ ...\ 1}_{I_{VK}+1} \end{bmatrix}^T \tag{11}$$

IV. The Practical Number of Vulnerable Bits for the Design of Physical-Layer Key Encoding

The analysis given in the last section is based on the assumption that the encoder knows I_{VK}. However, it is still unclear how we can derive I_{VK} in practice. We therefore propose two models of the enemy cryptanalyst, on which our estimate of I_{VK} will be based.

According to Fig. 4 (a), the enemy is modeled to have almost the same structure as the legitimate transmitter and receiver. However, since it can only estimate channel coefficients from an imaginary channel which differs from the key-generating channel used by the legitimate terminals, we propose an equivalent model in Fig. 4 (b). In the equivalent model, the enemy does estimate the key-generating channel, but the quantized key $\mathbf{K}$ is distorted by a binary symmetric channel (BSC), erring each estimated key bit with a probability p. The relationship between p and the I_{VK} previously mentioned is shown in Theorem 4.

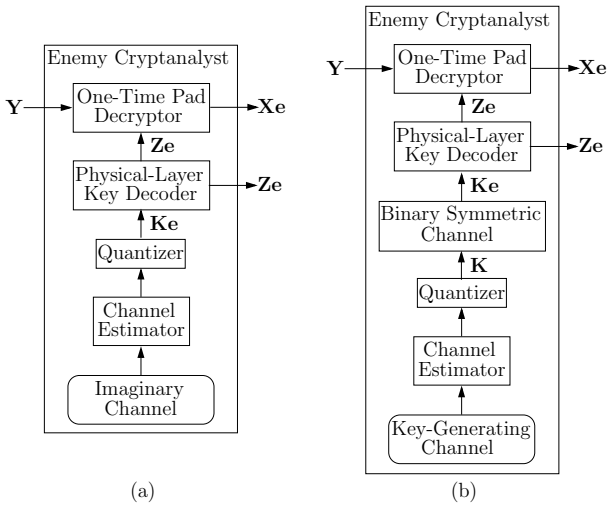

Fig. 4. The model of an enemy cryptanalyst (a), and its equivalent (b)

Theorem 4: If the enemy cryptanalyst behaves according to the model in Fig. 4 (b) having p as the error probability of the binary symmetric channel, and the generator matrix prototype in the form of equations (10) or (11) is used, when $I_{VK}+1$ is even or odd, respectively, the following conditions on I_{VK} are sufficient for perfect secrecy.

4.1. If $I_{VK}+1$ is even,

$$I_{VK}+1 \geq \begin{cases} \lceil -\frac{2}{\log_2(1-p)} \rceil , & \text{if } \lceil -\frac{2}{\log_2(1-p)} \rceil \text{ is even} \\ \lceil -\frac{2}{\log_2(1-p)} \rceil + 1 , & \text{if } \lceil -\frac{2}{\log_2(1-p)} \rceil \text{ is odd.} \end{cases} \quad (12)$$

4.2. If $I_{VK}+1$ is odd,

$$I_{VK}+1 \geq \begin{cases} \lceil -\frac{2}{\log_2(1-p)} \rceil - 1 , & \text{if } \lceil -\frac{2}{\log_2(1-p)} \rceil \text{ is even} \\ \lceil -\frac{2}{\log_2(1-p)} \rceil , & \text{if } \lceil -\frac{2}{\log_2(1-p)} \rceil \text{ is odd,} \end{cases} \quad (13)$$

where $\lceil -\frac{2}{\log_2(1-p)} \rceil$ is the smallest integer that is larger than $-\frac{2}{\log_2(1-p)}$.

Proof For perfect secrecy, the probability that the enemy can successfully decrypt x bits in the plaintext is at most $(\frac{1}{2})^x$, which equals the success probability of clueless guess. Therefore, according to equations (10) or (11), if $x = 1$,

$$(1-p)^{I_{VK}+1} \leq \frac{1}{2} \quad (14)$$

$$I_{VK}+1 \geq -\frac{1}{\log_2(1-p)} \quad (15)$$

Now, if $x = n$, which is the number of bits in the generated codeword as well as the number of bits in the plaintext, and $I_{VK}+1$ is even,

$$(1-p)^{\frac{n+1}{2}(I_{VK}+1)} \leq (\frac{1}{2})^n \quad (16)$$

For very large n, (16) becomes

$$\lim_{n\to\infty} (1-p)^{\frac{n+1}{2}(I_{VK}+1)} \leq \lim_{n\to\infty} (\frac{1}{2})^n \quad (17)$$

$$(1-p)^{\frac{n}{2}(I_{VK}+1)} \leq (\frac{1}{2})^n \quad (18)$$

$$I_{VK}+1 \geq -\frac{2}{\log_2(1-p)} \quad (19)$$

We can see that, as n increases, the minimal value of $I_{VK}+1$ that should be set also increases. If we denote by $\lceil -\frac{2}{\log_2(1-p)} \rceil$ the smallest integer that is larger than $-\frac{2}{\log_2(1-p)}$, the direct consequence of (19), when $I_{VK}+1$ is constrained to be an even number, will be the condition 4.1.

In case $I_{VK}+1$ is odd, we use the prototype in Eq. (11) and follow the same reasoning.

$$(1-p)^{\frac{n+1}{2}(I_{VK}+1)+\frac{n-1}{2}} \leq (\frac{1}{2})^n \quad (20)$$

$$\lim_{n\to\infty} (1-p)^{\frac{n+1}{2}(I_{VK}+1)+\frac{n-1}{2}} \leq \lim_{n\to\infty} (\frac{1}{2})^n \quad (21)$$

$$(1-p)^{\frac{n}{2}(I_{VK}+2)} \leq (\frac{1}{2})^n \quad (22)$$

$$I_{VK}+1 \geq -\frac{2}{\log_2(1-p)} - 1 \quad (23)$$

This results in the condition 4.2. Therefore, with the same p, we can set $I_{VK}+1$ to be smaller by 1 bit if it is odd than if it is even.

V. Discussion and Conclusion

We have proposed a secure protocol to ensure that the legitimate transmitter and receiver generate their secret key from the same physical channel. We have also included physical-layer key encoding into the system to provide perfect secrecy even when some key symbols are correctly estimated by the eavesdropper who knows the code.

We have suggested two simple generating matrix prototypes for our physical-layer key encoding for two specific cases, when $I_{VK}+1$ is even and when it is odd, where I_{VK} is the number of vulnerable key bits. $I_{VK}+1$ is related to I_K, the number of original key bits needed from the quantizer, by Theorem 2. In case $I_{VK}+1$ is unknown, we may use Theorem 4 to derive it from the probability p that the eavesdropper incorrectly estimates a key bit. For example, $I_{VK}+1$ is at least 5 when p is 0.25, yielding an asymptotic code rate of 3, as predicted by Theorem 3.

Acknowledgment

This work is supported by the German National Science Foundation (Deutsche Forschungsgemeinschaft, DFG)- Grant No. HE3654/11-1 "Unequal error protection and security approaches in wireless and network coding a study of continuous and discrete number designs" and the Bulgarian National Science Fund- Grant No. DO-02-135/2008 "Research on Cross Layer Optimization of Telecommunication Resource Allocation."

References

[1] J.L. Massey, "An Introduction to Contemporary Cryptology," *Proc. IEEE*, vol. 76, no. 5, pp. 533-549, May 1988.

[2] J. Wallace, "Secure Physical Layer Key Generation Schemes: Performance amd Information Theoretic Limits," *IEEE Int. Conf. Communications*, Dresden, Jun. 2009.

[3] A. Sayeed and A. Perrig, "Secure Wireless Communications: Secret Keys Through Multipath," *Proc. 2008 IEEE Int. Conf. Acoustics, Apeech, and Signal Processing*, Las Vegas, Mar.-Apr. 2008.

[4] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless Secret Key Generation Exploiting Reactance-Domain Scalar Response of Multipath Fading Channels," *IEEE Trans. Antennas and Propagation*, vol. 53, no. 11, pp. 3776-3784, Nov. 2005.

[5] M. Bloch, J. Barros, M.R.D. Rodrigues, and S.W. McLaughlin, "Wireless Information-Theoretic Security," *IEEE Trans. Information Theory*, vol. 54, no. 6, pp. 2515-2534, Jun. 2008.

[6] A. Sayeed, A. Perrig, "Secure Wireless Communications: Secret Keys Through Multipath," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Mar.-Apr. 2008.

[7] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, Nov. 1979.

[8] N. Cai, and R.W. Yeung, "Secure Network Coding," *Int. Symp. Information Theory*, Jun. 2002.