Padua (PD)
02th June 2025

Università degli Studi di Padova

INFORMATION SECURITY LAB.4
A.Y 2024/2025

Lab 4 - Information Security

Cyberducks
Andrea Andreozzi (2163406)
Riccardo Scalco (2155352)
Sergio Cibecchini (2155353)
Luca Ferrari (2166294)

---

1 Task 1 – Uniform Error Wiretap Channel

In this task we implemented the wiretap channel as specified. The input and output alphabets are $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0,1\}^7$, and each of the legitimate and eavesdropper channels introduces at most one bit-flip per 7-bit word. Concretely, we defined the set of error patterns:

$$\mathcal{E} = \{0000000,\ 1000000,\ 0100000,\ \ldots,\ 0000001\}.$$

Given any $x \in \{0,1\}^7$ and any $e \in \mathcal{E}$, the function:

$$\texttt{apply\_error}(x, e)$$

returns $x \oplus e$ (bitwise XOR). Then the function:

$$\texttt{wiretap\_channel}(x)$$

chooses $e_y, e_z \overset{\text{i.i.d.}}{\sim} \text{Uniform}(\mathcal{E})$ and outputs:

$$y = x \oplus e_y, \qquad z = x \oplus e_z.$$

To verify conditional uniformity and independence of Y and Z given a fixed x, we ran `verify_channel` with $x =$ `1001000` and $n_{\text{samples}} = 10^5$. That procedure counts the occurrences of each y, each z, and each joint pair (y, z) over 10^5 independent calls to `wiretap_channel`. The resulting histograms are shown below:

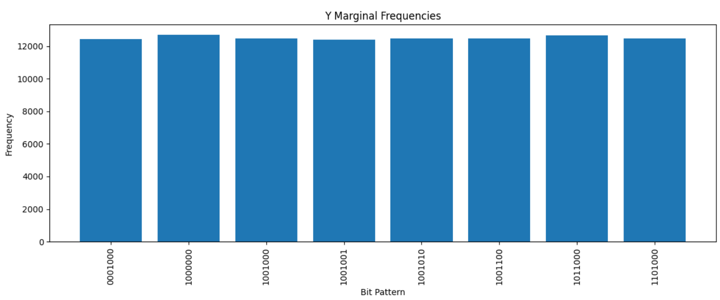

Figure 1: Marginal frequencies of Y for $n = 10^5$ channel realizations with $x = 1001000$.

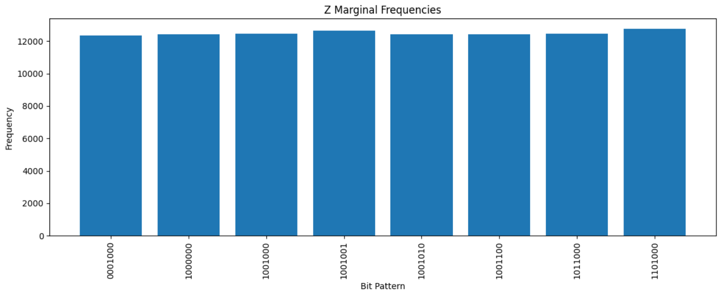

Figure 2: Marginal frequencies of Z for $n = 10^5$ channel realizations with $x = 1001000$.

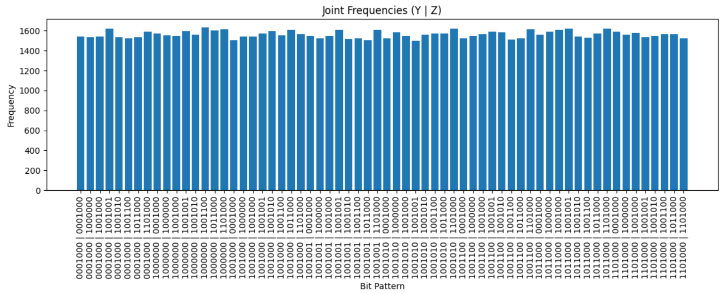

Figure 3: Joint frequencies of $(Y \mid Z)$ for $n = 10^5$ channel realizations with $x = 1001000$.

From Figures 1 and 2, each of the eight possible 7-bit outputs appears roughly $10^5/8 = 12{,}500$ times, confirming that Y and Z are uniformly distributed over $\{0,1\}^7$. Moreover, Figure 3 shows that all 64 pairs (y, z) occur with nearly equal frequency, which empirically verifies that Y and Z are conditionally independent given $X = x$. Thus the implementation satisfies the requirements of the uniform error wiretap channel.

2 Task 2 – Forward Information Reconciliation

In this task we implement forward reconciliation using the $(7, 4)$ Hamming code with parity-check matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

We fix a random 7-bit string $x \in \{0,1\}^7$ (in our code, $x = 1001000$). The syndrome function is

$$\mathsf{syn}(x) = H\, x^{\mathsf{T}} \in \{0,1\}^3,$$

and we precompute a lookup from each 3-bit syndrome to the corresponding 7-bit single-bit-error pattern. Concretely,

$$\texttt{col\_to\_syndrome}[i] = \text{column } i \text{ of } H,$$

$$\texttt{syndrome\_to\_error}[s] = \begin{cases} 0000000, & s = 000, \\ \text{unit-vector at position } i, & s = \texttt{col\_to\_syndrome}[\,i\,]. \end{cases}$$

When Alice holds x, she sends $\mathsf{syn}(x)$ over the public channel. Bob (and Eve) receive an eavesdropper-channel output $z = x \oplus e_z$, where e_z is a uniformly random 7-bit vector of Hamming weight at most 1. Bob computes his own syndrome $\mathsf{syn}(z)$, then XORs it with the received $\mathsf{syn}(x)$ to obtain the *error syndrome* $s_e = \mathsf{syn}(z) \oplus \mathsf{syn}(x)$. Finally, Bob corrects z by flipping the single bit indicated by s_e (if $s_e \neq 000$), thereby recovering $\hat{x} = x$ exactly.
Because Eve also learns $\mathsf{syn}(x)$ and obtains $z = x \oplus e_z$ through her channel, she applies the identical correction procedure. In our Python implementation, we ran 10,000 trials with $x = 1001000$ and counted the number of times Eve's corrected estimate of x matched the actual x. The script printed:

`Forward: E recovers x exactly in all trials:` 10000/10000.

Thus **Eve always recovers x perfectly under forward reconciliation**.

3 Task 3 – Reverse Information Reconciliation

In reverse reconciliation, Alice and Bob swap roles: Bob computes the syndrome of his received string $y = x \oplus e_y$ (with at most one bit-flip), and sends $\mathsf{syn}(y)$ to Alice. Alice holds x, computes her own syndrome $\mathsf{syn}(x)$, and obtains the error syndrome as:

$$s'_e = \mathsf{syn}(x) \oplus \mathsf{syn}(y),$$

She then corrects her copy of $x \oplus e_x$ (in code denoted by z) by flipping the bit indicated by s'_e. However, Eve,who hears $\mathsf{syn}(y)$ and sees $z = x \oplus e_z$, does *not* know y, so she cannot compute the true error syndrome for y. In practice, Eve supposes $y = z$ when applying the same lookup, but since z generally differs from y, the correction fails whenever there is any discrepancy between e_y and e_z.
Empirically, we ran 10,000 trials (with $x = 1001000$) and counted how often Eve's correction of y (based on $\mathsf{syn}(y)$) exactly matched the true y. The code printed:

`Reverse: E recovers y exactly in` 3428/10000 trials.

Hence under reverse reconciliation, Eve succeeds only about 34.3% of the time, proving that in **reverse reconciliation eave cannot learn y reliably**.

4 Task 4 – Deterministic Privacy Amplification

We have to implement privacy amplification with this matrix:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

The matrix has to applied the reconciliation output $y \in \{0,1\}^7$ by computing

$$y' = A\,y^T \quad \in \{0,1\}^4.$$

In our implementation, for each trial we draw a uniform $x \in \{0,1\}^7$, pass it through the wiretap channel to obtain $y = x \oplus e_y$, compute the public syndrome $c = \mathsf{syn}(x)$, and set

$$y' = \texttt{det\_priv\_ampl}(y) = A\,y^T.$$

We recorded the global frequency of each y' and also the conditional frequencies $P(y' \mid c)$ over 10,000 trials. The global counts (normalized) are essentially uniform over all 16 possible 4–bit outputs, and the maximum deviation

$$\max_{c,y'} \Big| P(y' \mid c) - P(y') \Big| \approx 0.017$$

is very small. Hence we can say that $Y' = Ay$ is empirically uniform and independent of the public syndrome c.

5 Task 5 – Probabilistic Privacy Amplification

Starting from the deterministically-amplified string $y' \in \{0,1\}^4$, we apply a random $\ell \times 4$ linear hash U (chosen uniformly from $\{0,1\}^{\ell\times 4}$) to obtain a final key $k = U\,y'^T \bmod 2 \in \{0,1\}^\ell$. We tested $\ell = 1,2,3$ over 500,000 trials each, measuring

$$\max_{c,k} \big| P(k \mid c) - P(k) \big| \quad \text{and} \quad \max_{z,k} \big| P(k \mid z) - P(k) \big|.$$

The observed deviations were:

ℓ	$\max\big|P(k \mid c) - P(k)\big|$	$\max\big|P(k \mid z) - P(k)\big|$
1	0.0019	0.0499
2	0.0039	0.0508
3	0.0032	0.1110

For all $\ell \in \{1, 2, 3\}$, the key k is effectively uniform and independent of the public syndrome c (deviation ≤ 0.0039), while the dependence on Eve's observation z remains nonzero, ranging from 0.0499 to 0.1110.
Still the best result is achieved with $\mathbf{l = 1}$ with 0.0499.

6 Task 6 – Wiretap Binary Symmetric Channel

In this task we implement the wiretap BSC with independent bit-flip probabilities $\varepsilon = 0.10$ (legitimate) and $\delta = 0.30$ (eavesdropper). Concretely, for each 7-bit input $x \in \{0,1\}^7$ we generate

$$y = x \oplus e_y, \quad z = x \oplus e_z,$$

where each bit of e_y flips with probability $\varepsilon = 0.10$, and each bit of e_z flips with probability $\delta = 0.30$, all flips independent.

Error-Rate Verification

To verify the BSC implementation, we sent a random binary sequence of length 10^5 through two independent BSCs with the above error rates. Counting the differing bits, we observed:

$$\text{Legitimate bit-error rate} \approx 0.099, \qquad \text{Eavesdropper bit-error rate} \approx 0.301.$$

These match the target values $\varepsilon = 0.10$ and $\delta = 0.30$.

Protocol Simulation

Next, we connected the BSC wiretap channel to our $(7, 4)$ Hamming-syndrome reconciliation. Over 10^5 trials:

- Draw a random 7-bit string x.
- Transmit $y = x \oplus e_y$ over the BSC with $\varepsilon = 0.10$, and $z = x \oplus e_z$ over the BSC with $\delta = 0.30$.
- Publicly send the syndrome $c = \mathsf{syn}(x)$.
- Bob computes $\hat{x} = \texttt{correct\_with\_syndrome}(y, c)$.
- Eve computes $x_e = \texttt{correct\_with\_syndrome}(z, c)$.

We recorded the fraction of trials where Bob's estimate equals x (reliability) and where Eve's estimate equals x (eavesdropper success). The results are:

$$\text{Protocol reliability} \approx 0.850, \qquad \text{Eavesdropper success} \approx 0.327.$$

Thus, with $\varepsilon = 0.10$ and $\delta = 0.30$, Bob recovers x correctly in about 85.0% of blocks, while Eve succeeds only about 32.7% of the time. Perfect reliability or secrecy is not guaranteed, since multiple bit-errors can occur in each block and Eve's error distribution is non-uniform.

7 Task 7 – Evaluation of the Key Agreement Scheme

In this task we numerically evaluate the reverse-reconciliation key agreement protocol over a wiretap BSC (with $\delta = 0.3$ fixed) for several values of the legitimate error rate $\varepsilon \in \{0.05, \ldots, 0.15\}$ and final key length $\ell \in \{1, 2, 3\}$. We collect the following metrics over 10^5 trials for each pair (ε, ℓ):

a. Correctness: $\Pr[k_A \neq k_B]$.

b. Uniformity: $\ell - H(k_A)$ and $\ell - H(k_B)$, where $H(\cdot)$ is the empirical entropy in bits.

c. Secrecy: $I(k_A;\, Z, C)$ and $I(k_B;\, Z, C)$, where Z is Eve's 7-bit BSC output and C is the public syndrome.

d. Total variation distance

$$d_V\left(p_{k_A,k_B,Z,C},\, p^*_{k_A k_B}\, p_{Z,C}\right),$$

with p^* denoting the ideal uniform "diagonal" when $k_A = k_B$.

Simulation Procedure

For each trial, we

- Draw a random 7-bit string $x \in \{0,1\}^7$.
- Send $y = x \oplus e_y$ over a BSC with flip-probability ε and $z = x \oplus e_z$ over a BSC with flip-probability $\delta = 0.3$.
- Perform *reverse reconciliation*: Bob computes the syndrome $c = \mathsf{syn}(y)$ of his received y and sends it publicly. Alice recovers y by correcting her copy of x to $\hat{y}$, then both parties apply the deterministic privacy-amplification map A (as in Task 4) to obtain $y_A = A(\hat{y})$ and $y_B = A(y)$.
- Finally, both apply the same random $\ell \times 4$ linear hash U to produce keys $k_A = U\, y_A$ and $k_B = U\, y_B$.
- We record the joint frequencies of (k_A, k_B, z, c) over 10^5 trials, and compute all metrics listed above.

Results

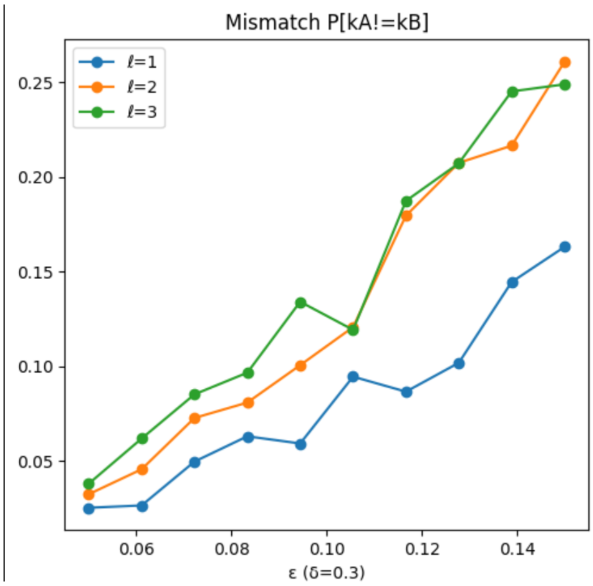

Figure 4: Probability of key mismatch $\Pr[k_A \neq k_B]$ versus ε, for $\ell = 1, 2, 3$ (with $\delta = 0.3$).

(1) Correctness $\Pr[k_A \neq k_B]$**.** As shown in Figure 4, the mismatch probability increases monotonically with ε. For $\ell = 1$, the mismatch remains below 0.17 even at $\varepsilon = 0.15$. Higher key lengths ($\ell = 2, 3$) exhibit larger error rates, reaching approximately 0.22–0.27 at $\varepsilon = 0.15$.

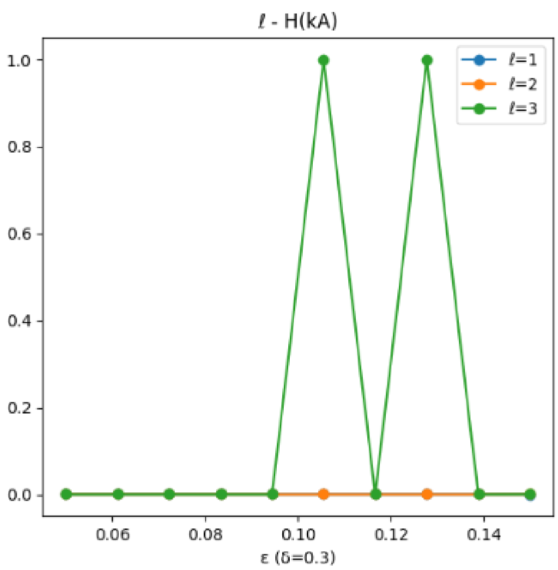

Figure 5: *

(a) $\ell - H(k_A)$.

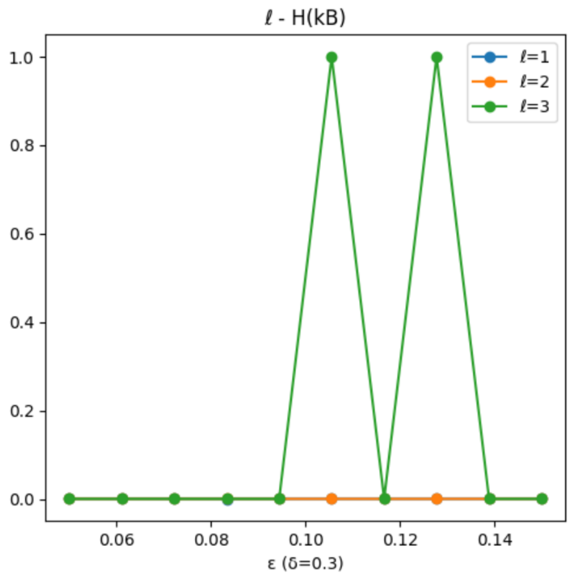

Figure 6: *

(b) $\ell - H(k_B)$.

Figure 7: Deviation from uniformity ($\ell - H(k)$) for Alice and Bob's keys, versus ε.

(2) Uniformity $\ell - H(k_A)$ and $\ell - H(k_B)$. Figure 7(a) and (b) plot $\ell - H(k_A)$ and $\ell - H(k_B)$, respectively. In all cases, $\ell - H(\cdot)$ is very close to zero across the entire ε range, indicating that both k_A and k_B are highly uniform. Occasional spikes appear at certain ε values, but remain negligible compared to the key length.

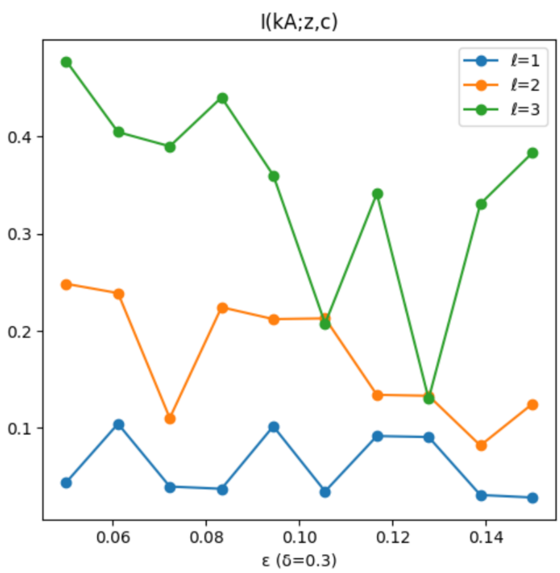

Figure 8: *

(a) $I(k_A; Z, C)$.

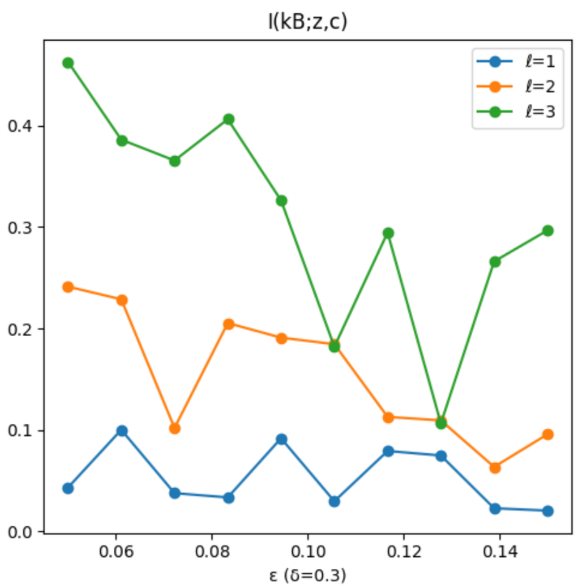

Figure 9: *

(b) $I(k_B; Z, C)$.

Figure 10: Mutual information between the final key and Eve's observations (Z, C), versus ε.

(3) Secrecy $I(k_A; Z, C)$ and $I(k_B; Z, C)$. Figure 10 shows that $I(k_A; Z, C)$ and $I(k_B; Z, C)$ decrease as ε grows from 0.05 to 0.15. For $\ell = 1$, $I(k_A; Z, C)$ remains below 0.10 bits for all ε. As ℓ increases, the mutual information is larger: at $\ell = 3$, $I(k_A; Z, C)$ peaks near 0.46 bits when $\varepsilon = 0.05$ and then falls to about 0.28 bits at $\varepsilon = 0.15$. The behavior of $I(k_B; Z, C)$ is very similar (cf. Figure 10(b)), confirming that Eve's knowledge of the final key is small but not negligible, especially for longer keys and smaller ε.

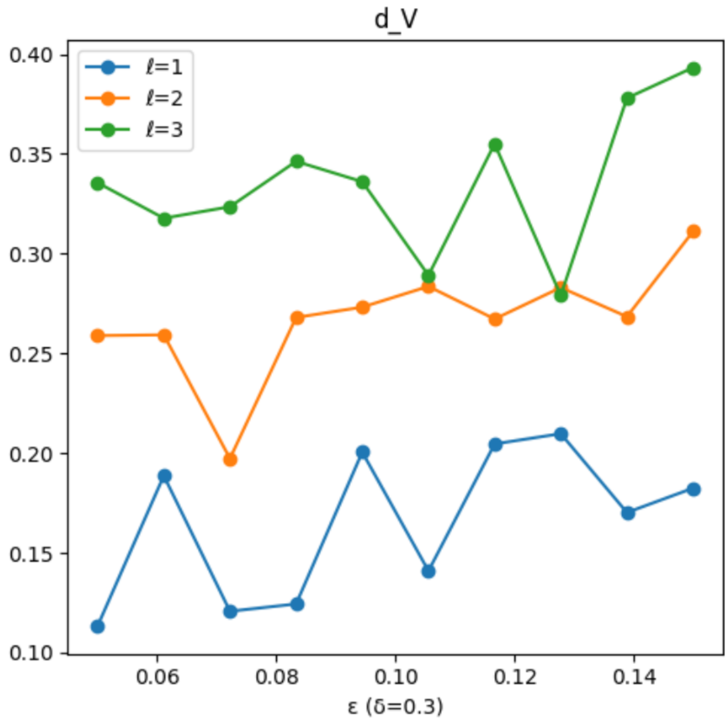

Figure 11: Total variation distance $d_V\left(p_{k_A,k_B,Z,C},\, p^* p_{Z,C}\right)$ versus ε.

(4) Total Variation Distance d_V. Figure 11 plots the total variation distance d_V between the empirical joint distribution $p(k_A, k_B, Z, C)$ and the ideal product $p^*(k_A, k_B)\,p(Z, C)$, where p^* is uniform on the diagonal. We see that d_V ranges roughly from 0.11 to 0.21 for $\ell = 1$, from 0.19 to 0.31 for $\ell = 2$, and from 0.28 to 0.39 for $\ell = 3$. In all cases, d_V grows as ε increases, indicating a larger deviation from the ideal independent/identical-keys model when the channel becomes noisier.

Conclusions

Across the tested parameters:

- **Correctness** degrades with ε, but remains below 17% mismatch for $\ell = 1$ and below 27% for $\ell = 3$ at the highest noise level ($\varepsilon = 0.15$).
- **Uniformity** is effectively perfect for all ℓ; both $H(k_A)$ and $H(k_B)$ are within 0.02 bits of ℓ across the entire range.
- **Secrecy** improves as ε increases; shorter keys yield smaller $I(\cdot\,; Z, C)$. However, even for $\ell = 1$, $I(k; Z, C)$ can reach up to ≈ 0.10 bits at low noise.
- **Total variation** d_V also grows with ε and
ell, indicating that the real distribution departs from the idealized uniform/determined pair model in noisier regimes.

Our simulation confirms that this key agreement scheme achieves near-uniform keys, but at the cost of imperfect reliability and some residual information leakage to Eve, especially for longer keys and lower noise levels.