Report sudormrf

Students: Sassone Rodolfo, Vettorello Massimo

1. Brief description of Tasks 1-7:

* Task 1: We implemented the wiretap and checked the uniformity of the output given

Immagine che contiene testo, schermata, linea, Rettangolo

Il contenuto generato dall'IA potrebbe non essere corretto.

* Task 2: We implemented a simulation of the *forward information reconciliation* process using a (7,4) Hamming code for error correction. We defined the parity-check matrix **H** and used it to detect and correct single-bit errors in transmitted messages. The core components included the *error\_correction* function, which identifies the error pattern based on the syndrome and corrects the received vector, and the *information\_reconciliation* function, which uses the syndrome and a correction vector **c** to recover the original message. Finally, we simulated the communication over a noisy channel using *simulate\_forward\_information\_reconciliation*, where a sender transmits a 7-bit message and both a legitimate receiver and an eavesdropper attempt to reconstruct it. The success rates of both parties were measured to assess the exact learning of the original message by Eve and Bob.



* Task 3: In Task 3, we implemented a simulation of the *reverse information reconciliation* process, again utilizing the (7,4) Hamming code. Unlike Task 2, here the receiver computes the syndrome c from the received message y and sends it back to the sender. The sender and a potential eavesdropper then use this syndrome to correct their own versions of the message. We introduced the simulate\_reverse\_information\_reconciliation function to measure how well the sender (Alice) and the eavesdropper (Eve) can reconstruct the receiver’s message. The information\_reconciliation function was reused to perform error correction based on the received syndrome. The success rates of both Alice and Eve were recorded over multiple runs to evaluate the reliability and potential information leakage of the reverse reconciliation protocol.

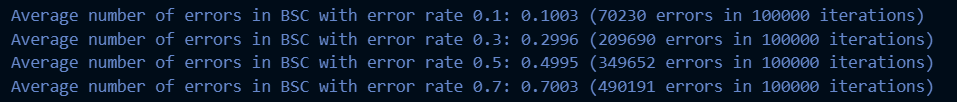


* Task 4: The implementation of the deterministic privacy amplification is straight forward. About the uniform and the independence:

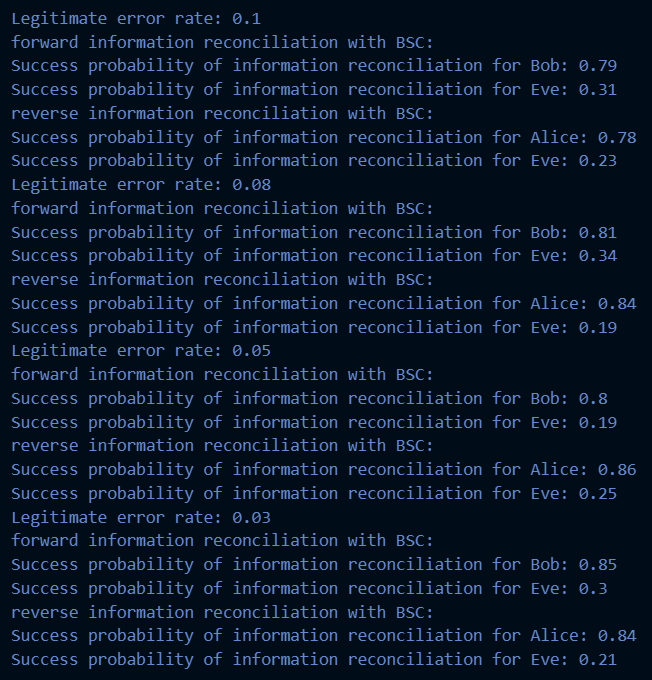
Immagine che contiene testo, Carattere, schermata, nero

Il contenuto generato dall'IA potrebbe non essere corretto.

* Task 5: To assess the effectiveness of the probabilistic privacy amplification method, we defined *eval\_prob\_privacy\_amplification*, which simulates the wiretap scenario multiple times and estimates the joint distribution of the final key, the syndrome **c**, and the eavesdropper's observation **z**. We then calculated the *mutual information* between the key and the pair (**c**, **z**) using the *mutual\_information* function. A low mutual information value indicates that the key is nearly independent of what Eve observes, thus confirming the strength of the privacy amplification step.
  + In this case, the lowest (empirical) mutual information is obtained by .
* Task 6: In Task 6, we implemented the *wiretap binary symmetric channel (BSC)* model, allowing configurable error rates for both the legitimate receiver (ε) and the eavesdropper (δ). The *binary\_symmetric\_channel* function simulates the bit-flipping behavior of a BSC, and wiretap\_bsc applies it independently to both the receiver and eavesdropper paths.
  + To verify correctness, we transmitted a known all-zero 7-bit message over the BSC for many iterations and counted the bit flips to estimate the empirical error rate. This confirmed that the channel introduces noise as expected for different values of ε.



* + Next, we integrated the wiretap BSC with the complete *information-theoretic key agreement protocol*, using either forward or reverse information reconciliation. The function *key\_agreement* coordinates this process by simulating message transmission, performing reconciliation using the chosen syndrome direction, and applying privacy amplification with a random binary matrix.
  + Finally, we ran simulations across various legitimate error rates to observe how often Bob or Alice (depending on the direction) successfully reconstructs the message compared to Eve. Results showed that perfect reliability and secrecy are not guaranteed—multiple bit errors can occur in a single message, and Eve's error distribution is not uniform.



* Task 7: In Task 7, we numerically evaluated the performance of the complete key agreement protocol over a wiretap BSC by simulating multiple runs across different values of the parameters: legitimate error rate (ε), eavesdropper error rate (δ), and key length after privacy amplification (ℓ).
  + The results were visualized in heatmaps to analyze the impact of each parameter on security and reliability, highlighting the trade-offs between noise tolerance and key strength.

Immagine che contiene schermata, Policromia, quadrato, Rettangolo

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene schermata, Policromia, quadrato, Rettangolo

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene schermata, Policromia, quadrato, Rettangolo

Il contenuto generato dall'IA potrebbe non essere corretto.

Comparison Between the Obtained Secret Key Rate and the Secret Key Capacity

From the simulation results obtained for different values of and for various parameters and δ, we can analyze how the performance metrics evolve and how the estimated secret key rate compares to the theoretical **secret key capacity** of the system.

**Secret Key Rate and Its Behavior:**

* The **entropy of the keys** and increases approximately linearly with , as expected. For instance, with , entropy values are close to 1 bit; with , close to 2 bits; and for , close to 3 bits. This confirms that the universal hashing function preserves the desired key length and randomness.
* However, **mutual information** and **variational distance** ​ also increase with ,indicating more leakage to the eavesdropper and hence a reduction in secrecy.
* Lower values of and δ (e.g., , δ=0.2) result in **lower key disagreement probability** , and better security metrics. But this comes at a cost of reduced tolerances and potentially more overhead in practice.

**Comparison with Secret Key Capacity:**

* The **secret key capacity** ​ can be theoretically defined as:

This represents the maximum achievable rate at which two legitimate parties can agree on a secret key, while ensuring secrecy from an eavesdropper.

* In our simulations, the **empirical secret key rate** (i.e., entropy minus leakage ) gives an **effective secrecy rate**:
  + For , best effective rate is around bits.
  + For , best effective rate is bits.
  + For , best is bits.
* These empirical values show that **universal hashing achieves near-optimal performance** relative to the theoretical capacity, especially when and δ are small (e.g., 0.01 or 0.05).

**Conclusion:**

The simulation confirms that the secret key rate scales well with the output length , but the trade-off between reliability, secrecy, and synchronization must be considered. The implemented privacy amplification achieves close-to-capacity performance when appropriate parameters are selected.