Secret Key Generation for FDD Systems Based on Complex-Valued Neural Network

Xinwei Zhang[1], Guyue Li[1,3], Zongyue Hou[1], Aiqun Hu[2,3]

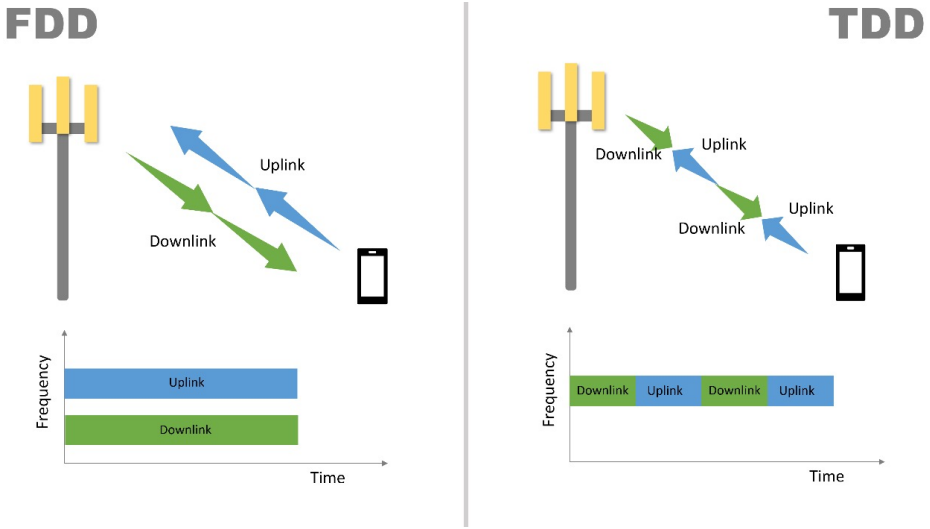
[1]School of Cyber Science and Engineering, Southeast University, Nanjing, China
[2]National Mobile Communications Research Laboratory, Southeast University, Nanjing, China
[3]Purple Mountain Laboratories for Network and Communication Security, Nanjing, China

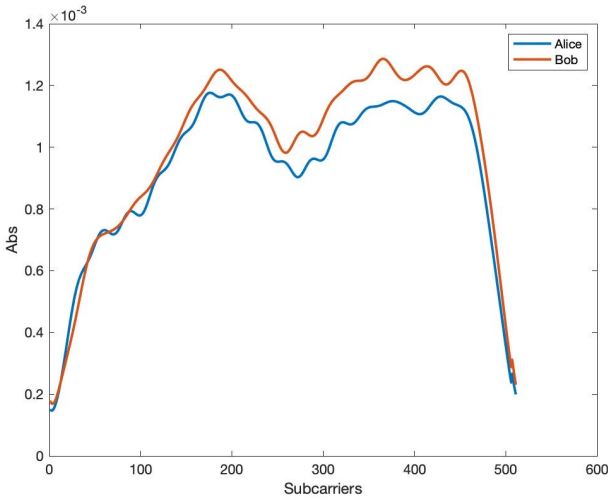
VTC2021-Fall

Directory

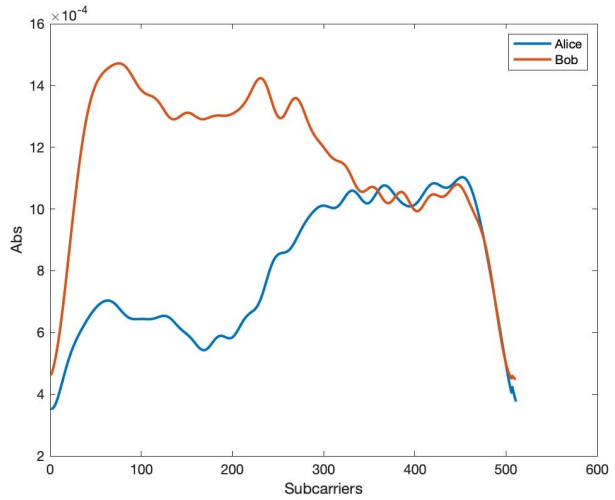
Background

- The security of the wireless air interface is important.
- Physical-layer secret key generation: a new type of highly secure and convenient key distribution method.
- This method depends on the channel reciprocity.
- Time division duplexing (TDD) & Frequency division duplexing (FDD)





TDD



FDD

Channel model and problem formulation

- Channel model: FDD-OFDM, SISO.
  - Channel frequency response (CFR):

$$H(f,n)=\sum_{l=0}^{L-1}\alpha_l e^{(-j2\pi f\tau_l+j\phi_l)}e^{(-j2\pi\tau_l f_n)}$$

$$\begin{cases}\mathbf{H}_{AB}=\{H(f_{AB},1),...,H(f_{AB},N)\}\\ \mathbf{H}_{BA}=\{H(f_{BA},1),...,H(f_{BA},N)\}\end{cases}$$

$$\mathbf{\Psi}_{f_{BA}\to f_{AB}}=\mathbf{H}_{BA}\to\mathbf{H}_{AB}.$$

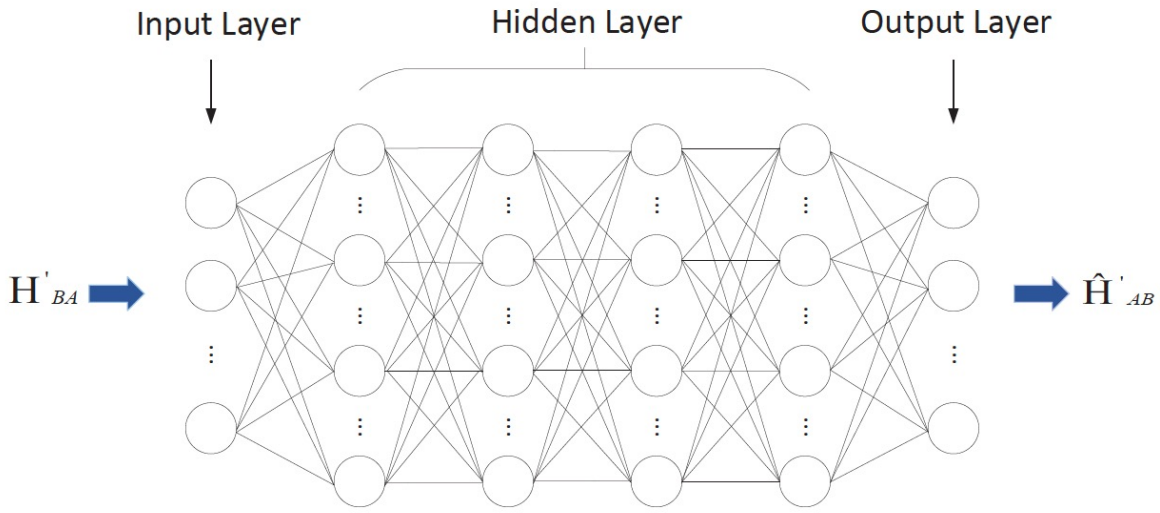
- A certain mapping function between channels in different frequency bands was proved in [1].
- Problem: how to get this mapping function and use it for FDD key generation.
- This paper proposed a Complex-Valued neural Network (CVNet) based secret key generation method.

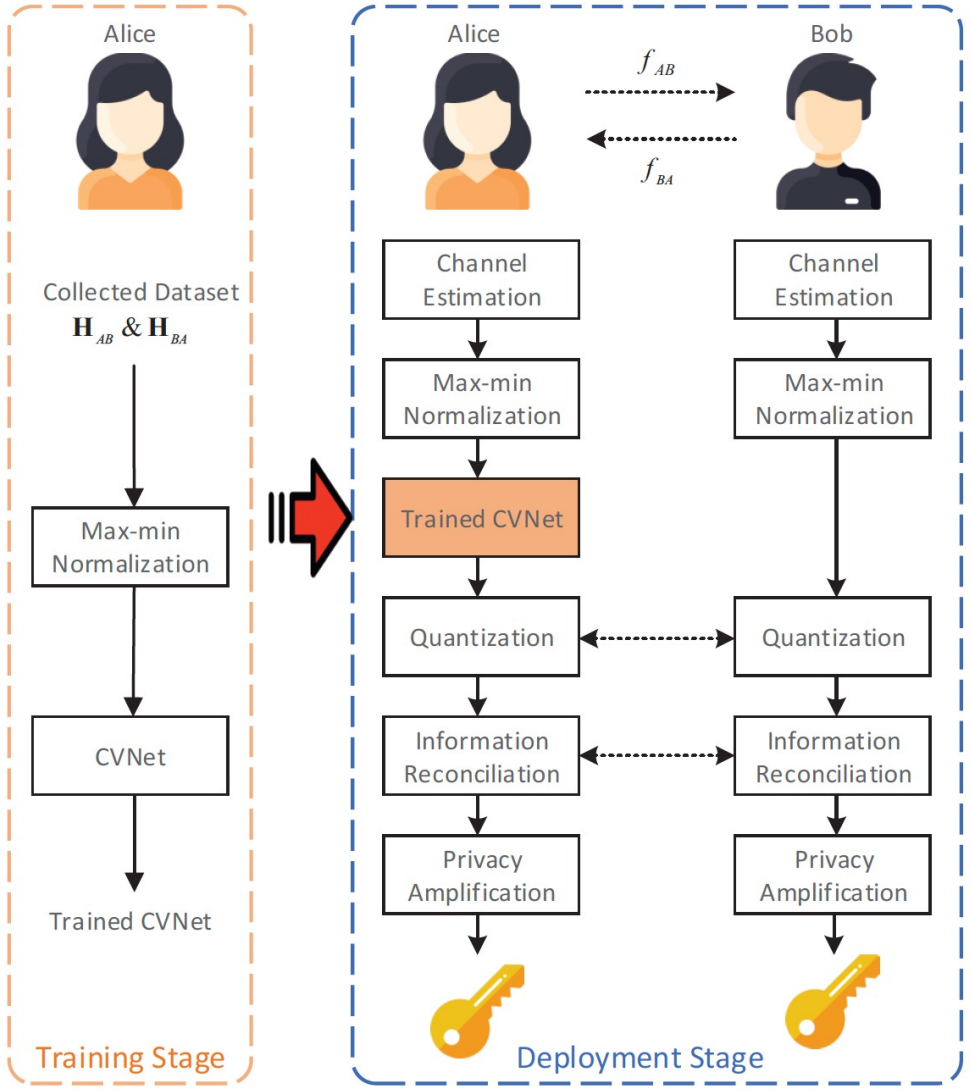
[1] X. Zhang, G. Li, J. Zhang, A. Hu, Z. Hou, and B. Xiao, "Deep Learning-based Physical-Layer Secret Key Generation for FDD Systems," accepted by IEEE Internet of Things Journal.

The CVNet based key generation scheme

- Two stages:
  Training stage and deployment stage
- Training stage:

$$Loss(\mathbf{\Omega}) = \frac{1}{N * M} \sum_{m=1}^{M} \| \hat{\mathbf{H}}_{AB}^{'(m)} - \mathbf{H}_{AB}^{'(m)} \|_2^2$$





The CVNet based key generation scheme

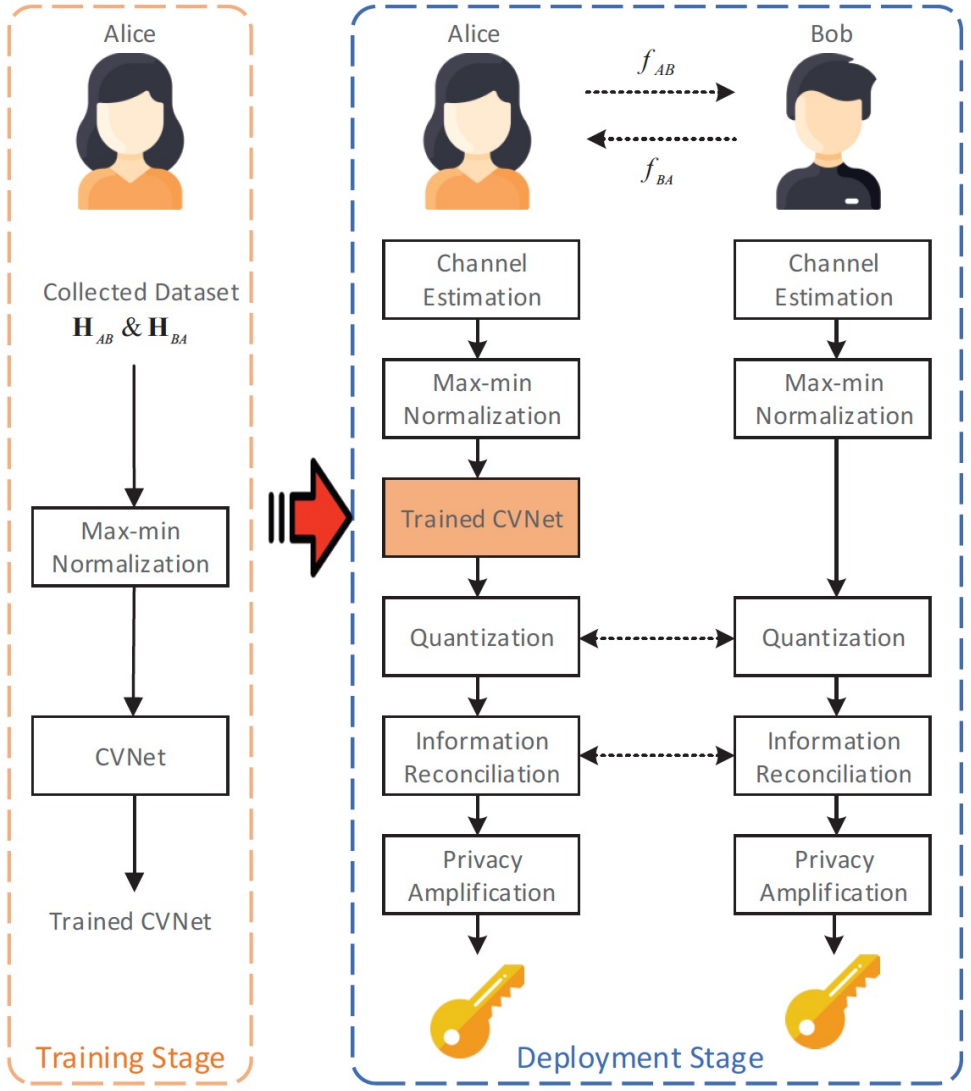
- Deployment stage
  - Quantization:

$$\mu = \frac{1}{2N}\sum_{n=1}^{N}(|\ \Re(H(f,n))\ | + |\ \Im(H(f,n))\ |)$$

$$\sigma^2 = \frac{1}{2N-1}\sum_{n=1}^{N}\{(|\ \Re(H(f,n))\ | -\mu)^2 + (|\ \Im(H(f,n))\ | -\mu)^2\}.$$

$$\eta_+ = \mu + \alpha \cdot \sigma$$

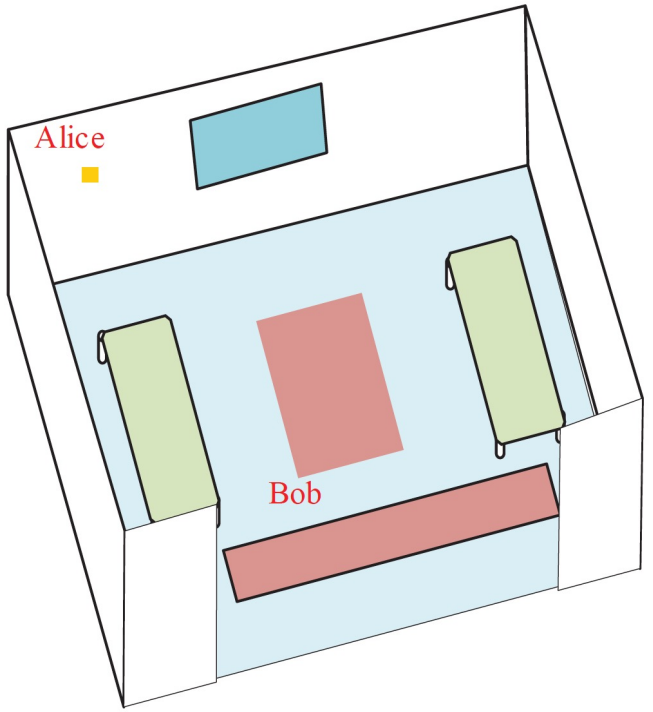
$$\eta_- = \mu - \alpha \cdot \sigma$$



Performance evaluation

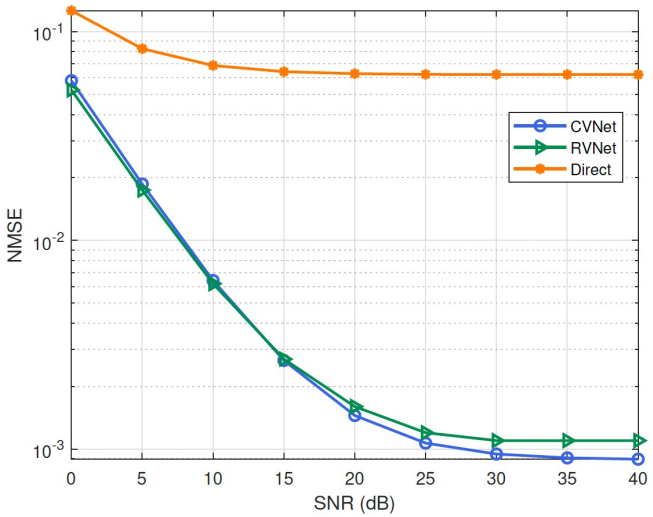
- Simulation setup:
  - Frequencies: 2.4 G and 2.5G
  - Dataset source: 3D ray-tracing simulator Wireless InSite.
- Keras with the TensorFlow backend as the deep learning framework.
- The CVNet is implemented on a workstation with one Nvidia GeForce GTX 1660Ti GPU.

Parameter	Value
Number of neurons in Input/Output layers	64
Number of neurons in hidden layers	(256,512,512,256)
Optimization	ADAM [23]
Learning rate	1e-3
Number of epochs	500
Batch size	128
Number of training samples	80,000
Number of testing samples	20,000

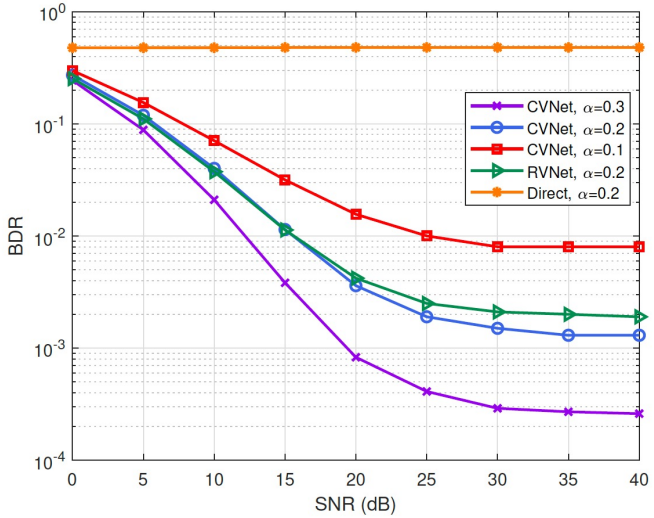


Performance evaluation

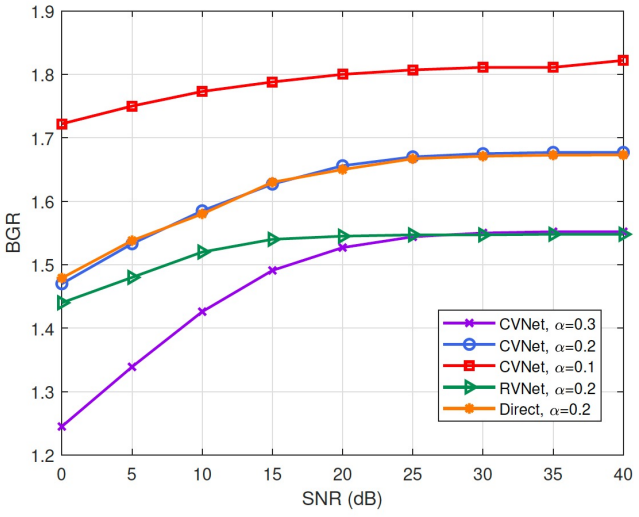
- Two benchmarks:
  - Direct method.
  - Real-valued neural network(RVNet) based key generation: hidden layer: (512, 1024, 1024, 512).
- **Result 1**: In the case of high SNRs, the CVNet based key generation method will be better than the RVNet based method.



Normalized Mean Square Error (NMSE)



Bit Disagreement Ratio (BDR)



Bit Generation Rate (BGR)

Performance evaluation

- Complexity analysis:
  - Floating point operations (FLOPs):
    - RVNet: $\sum_{r=1}^{R-1}(K_{r-1}+1)K_r$, 2,231,424
    - CVNet: $4\sum_{r=1}^{R-1}(K_{r-1}+1)K_r$, 2,234,624
  - The number of trained parameters:
    - RVNet: $\sum_{r=1}^{R-1}(K_{r-1}+1)K_r$, 1,117,312
    - CVNet: $2\sum_{r=1}^{R-1}(K_{r-1}+1)K_r$, 1,117,312

- **Result 2**: The FLOPs required for the CVNet and the RVNet are almost the same, while the trained parameters required by the CVNet are only half of those required by the RVNet.

Thank you!
(Q & A)

Contact Us: zxw1998@seu.edu.cn