Unification of Encryption and Coupling

Can They be Linked? How and in Which Context?

Outline

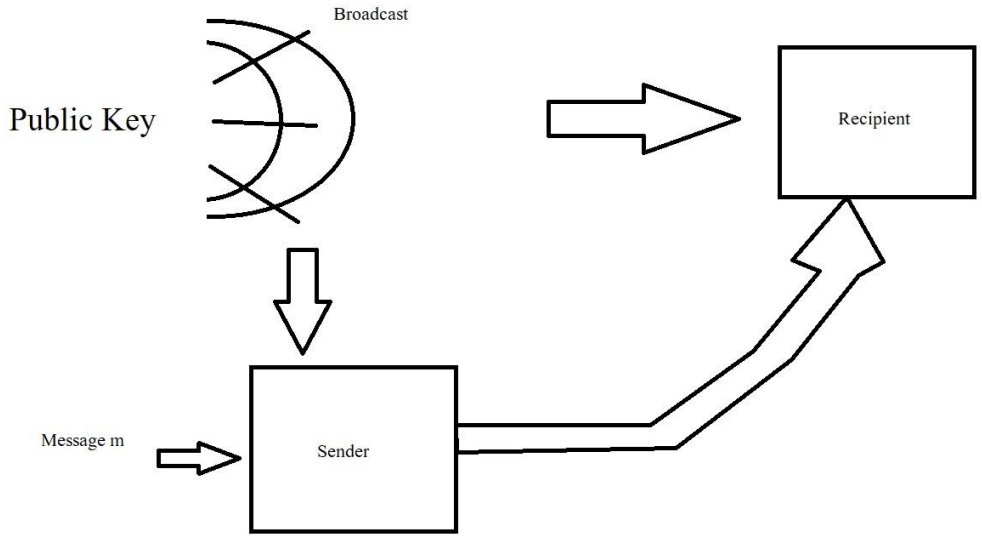
1. Asymmetric Encryption
2. Asymmetric Encryption and Decryption
3. RSA Algorithm
4. About RSA Algorithm
5. From RSA Towards FHE
6. Preview of Homomorphic Encryption
7. Ephaptic Coupling: The Setting
8. Ephaptic Coupling: RC Model
9. Ephaptic Coupling Review
10. MATLAB Code and Output: Two Synchronizing Axons
11. Perceptron Review
12. The Ephaptic Perceptron Model
13. MATLAB Code and Output: Two perceptrons, uncoupled (alpha = 0)
14. MATLAB Code and Output: Two perceptrons, coupled linearly (alpha = 1)
15. MATLAB Code and Output: Two perceptrons, coupled nonlinearly (alpha = 2)
16. MATLAB Code and Output: Two perceptrons, coupled nonlinearly (alpha = 3)
17. Observations
18. The Answer
19. Next Steps
20. Future Work Directions
21. References / Bibliography

Part 1: On Encryption

Asymmetric Encryption

- It involves the use of two different cryptographic keys: a public key and a private key.
- The public key is freely distributed and used for encryption. It can be shared with anyone, including the sender and any potential recipient. The public key is derived from the private key but cannot be used to determine the private key.
- The private key is kept secret and known only to the owner. It is used for decryption. The private key is mathematically related to the public key, but it cannot be easily derived from it

Asymmetric Encryption - contd.



Asymmetric Encryption and Decryption

Encryption:

1. The recipient generates a key pair: a public key and a private key.
2. The recipient shares the public key with anyone who wants to send them an encrypted message.
3. The sender obtains the recipient's public key.
4. The sender uses the recipient's public key to encrypt the message. This ensures that only the recipient can decrypt it.

Decryption:

1. The recipient uses their private key to decrypt the message. Since the private key is known only to the recipient, the message remains secure.
2. Once decrypted, the recipient can read the original message.

RSA Algorithm

1. Key Generation:
   - Select two distinct prime numbers, p and q.
   - Compute n = p * q, which is the modulus.
   - Compute φ(n) = (p - 1) * (q - 1), where φ is Euler's Totient function.
   - Choose an integer e (1 < e < φ(n)) such that e and φ(n) are coprime. e is the public exponent.
   - Compute d as the multiplicative inverse of e modulo φ(n), such that (d * e) mod φ(n) = 1. d is the private exponent.
   - The public key is (n, e), and the private key is (n, d).
2. Encryption:
   - Convert the plaintext message into a numerical representation (usually using a suitable encoding scheme like UTF-8 or ASCII).
   - Split the message into blocks if necessary.
   - Compute the ciphertext c for each block using the formula c = (m^e) mod n, where m is the numerical representation of the plaintext message.
3. Decryption:
   - Given the ciphertext c, compute the plaintext message m using the formula m = (c^d) mod n.

About RSA Algorithm

- It was invented by Rivest, Shamir and Adleman [10]
- It is a multiplicatively homomorphic encryption scheme
- To understand this term, we need to know some background
- In RSA the public key is denoted pk = (N,e) where N is an integer and so is e
- The plaintext (which needs to be encrypted) is operated upon by the RSA encryption algorithm, using the public key
- The result is the ciphertext
- Consider the RSA ciphertexts \psi_i
- We make the assignment \pi_i^{e} mod N to each \psi_i
- Here the \pi_i stand for the plaintexts
- Given pk and the \psi_i, one can efficiently compute the product of the ciphertexts \psi_i and find it to be equal to (product of \pi_i)^e mod N
- This is a multiplicative homomorphism between the ciphertexts and the plaintexts, a type of ``nice" mapping between the two
- Thus, by not even touching the plaintexts one has the ability to compute their product (modulo N)

From RSA Towards FHE

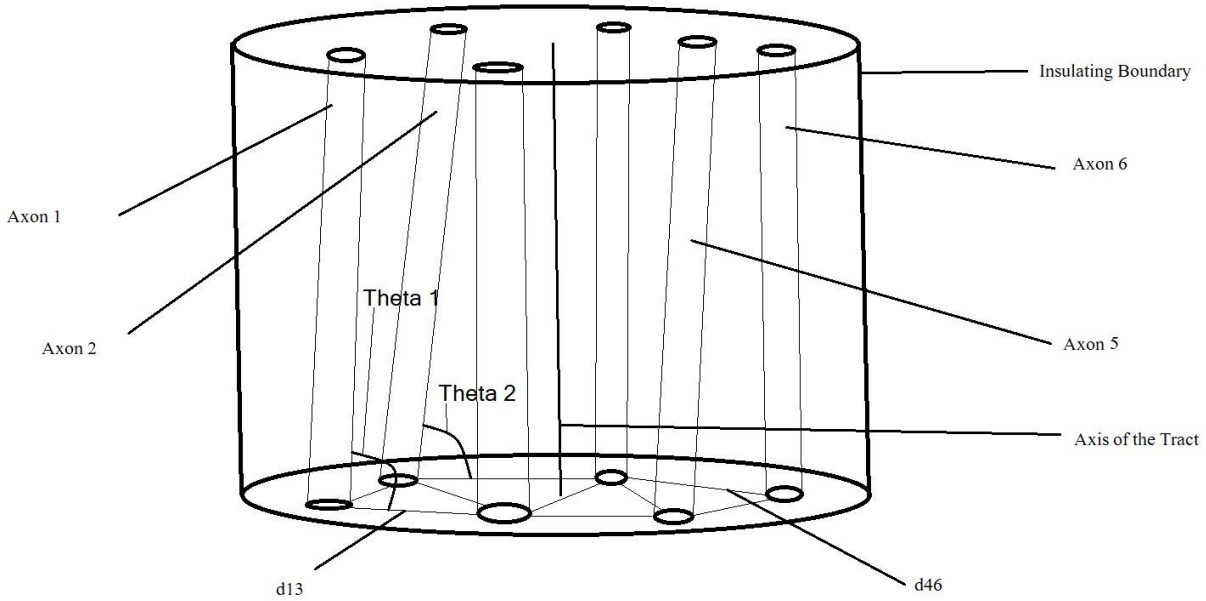
- We see that this ability to perform a homomorphic multiplication could have inspired RSA to propose fully homomorphic encryption where any and every operation can be performed homomorphically (Gentry, 2009)
- However, it was soon realized that this was a very difficult task
- It was only in the 21st century that Craig Gentry could solve it in his Stanford dissertation
- If we understand the RSA Algorithm better, it might help to understand all cryptosystems better
- In future we can work further towards understanding Gentry's scheme; the next slide offers a sneak preview

Preview of Homomorphic Encryption

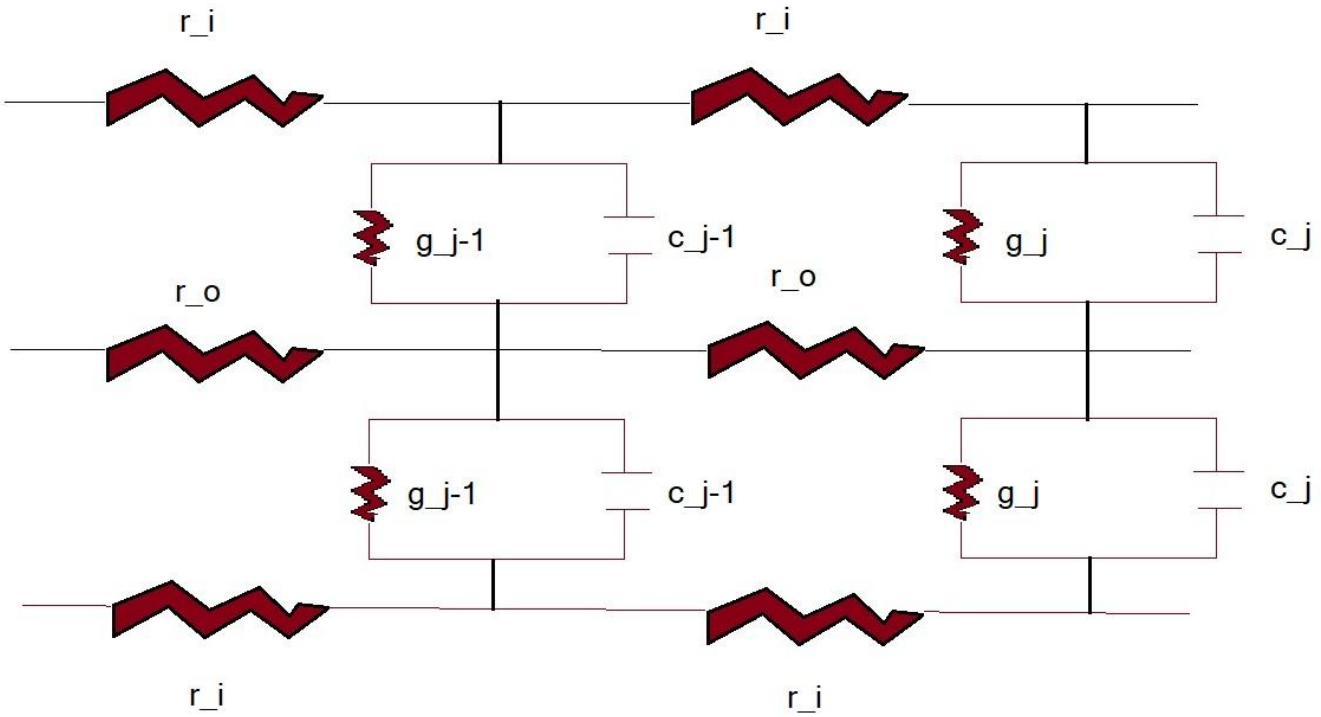
- A homomorphic public key encryption scheme E has four algorithms KeyGenE , EncryptE , DecryptE , and an additional algorithm EvaluateE that takes as input the public key pk, a circuit C from a permitted set CE of circuits, and a tuple of ciphertexts $\Psi = \langle \psi 1, \ldots, \psi t \rangle$; it outputs a ciphertext ψ.
- The computational complexity of all of these algorithms must be polynomial in security parameter λ and (in the case of EvaluateE) the size of C.
- E is correct for circuits in CE if, for any key-pair (sk, pk) output by KeyGenE (λ), any circuit $C \in CE$, any plaintexts $\pi 1, \ldots, \pi t$, and any ciphertexts $\Psi = \langle \psi 1, \ldots, \psi t \rangle$with $\psi i \leftarrow$ EncryptE (pk, πi), it is the case that:
- $\psi \leftarrow$ EvaluateE (pk, C, Ψ) $\Rightarrow$ $C(\pi 1, \ldots, \pi t)$ = DecryptE (sk, ψ)

Part 2: On Ephaptic Coupling

Ephaptic Coupling: The Setting



Ephaptic Coupling: RC Model

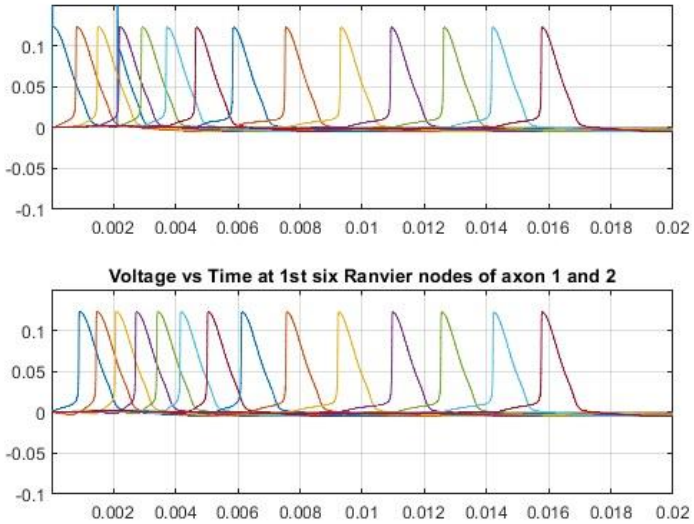


Ephaptic Coupling Review

- Form of axon-axon coupling
- Current in the extracellular space has a return path which couples neighboring axons; due to overall bundle insulation
- Results in synchronization between impulse trains
- Linear coupling studied in detail (2017: dissertation)
- Nonlinear coupling studied recently (2023: joint work with Suma Manjunath)

MATLAB Code and Output: Two Synchronizing Axons

Demo for N=2 case.
C:\Users\acer\Documents\MATLAB\CorrectedAxonGe2022Copy\mainaxonAug17_2020v7.m



Part 3: Of Perceptrons

Perceptron Review

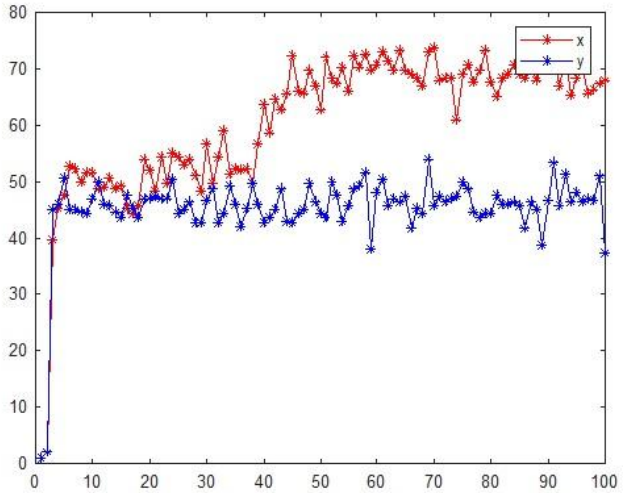
- Rosenblatt introduced the perceptron as a linear classifier in 1958
- Simple neuron model
- Dendritic inputs are multiplied by weights and combined with biases
- These intermediate values are summed together
- The summed output is passed through a signum or logarithmic or other activation function
- The output of the activation step is the neural output
- Decision boundary is linear for a single perceptron meaning that data is separated into classes C1 and C2, but there should be a hyperplane separating the two classes, not a complicated surface

The Ephaptic Perceptron Model

- The advantage of this model is that by changing the value of alpha (the power of the pre-fixed external voltage) one can go from an uncoupled linear system (when alpha is 0) to a coupled linear system (when alpha =1) to a coupled nonlinear system (when alpha > 1).
- One can therefore propose a single system that can do both linear as well as nonlinear classification tasks, when the parameter alpha is suitably set
- If the parameter is adaptively set based on an initial study of the problem, then it may be possible for the system itself to shift between linear and appropriate nonlinear classification based on the input training data

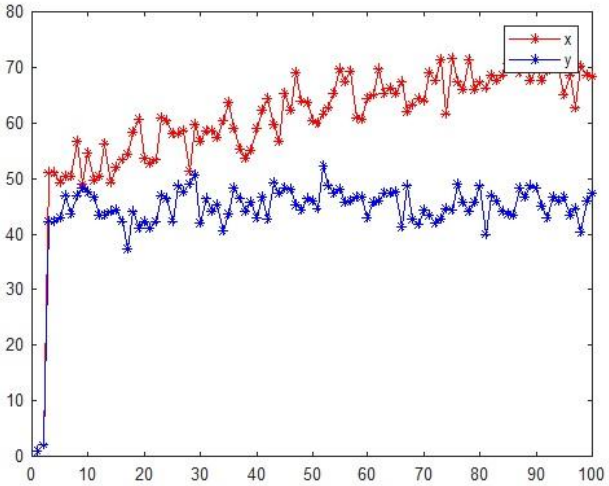
MATLAB code and output: Two perceptrons, uncoupled

Demo for alpha = 0.
C:\Users\acer\Documents\MATLAB\VectorEphapticPerceptronNonlinear\percepmainv7.m



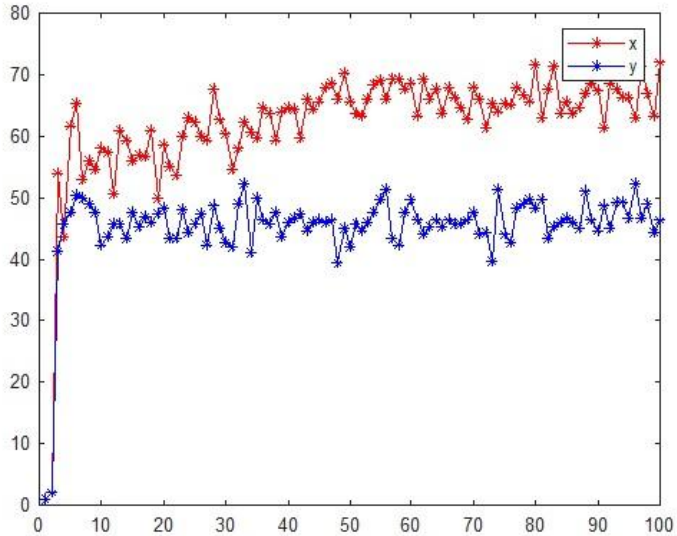
MATLAB code and output: Two perceptrons, coupled linearly

Demo for alpha = 1.



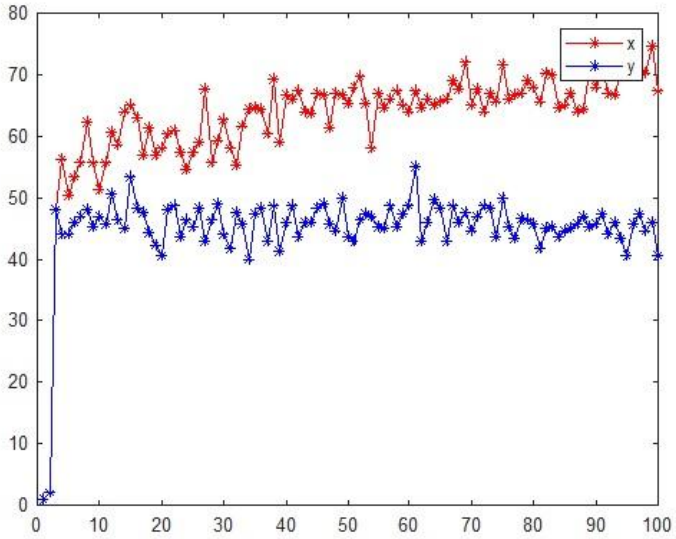
MATLAB code and output: Two perceptrons, coupled non-linearly

Demo for alpha = 2



MATLAB code and output: Two perceptrons, coupled non-linearly

Demo for alpha = 3



Observations

- All other parameters being the same, the coupled perceptrons give a more steady and early-start rise to the saturation value of 70 percent for the x-output, as compared to the uncoupled case.
- The y-output hovers at nearly 50 percent in all cases.
- There doesn’t seem to be much of an advantage from varying alpha beyond 1 (linear coupling).

Part 4: Final Notes

The Answer to the Subtitle

- In summary, we would use FHE in the rogue model for two purposes: assignment function distribution and slice security.
- And these rogues would be classified in two ways:
  - over time using the ephaptic perceptron which would learn the network's roguishness related behavior, and
  - post assignment-function-reconstruction when the neighbor's behavior would be adjudged as roguish or not based on an ability to abet the FHE scheme.
- Both are observation-based categorisations of node roguishness; the coupling based scheme gets better over time, whereas the FHE scheme is a one-shot categorisation applied repeatedly
- One could develop a meta-scheme that combines the responses of these two schemes in appropriate proportions, in order to get an improved performance
- Thus the answer is "yes", encryption and coupling may be linked and jointly used to defend against rogues.

Next Steps

- Test for higher dimensional data sets and complex boundaries
- Compare the various properties of these algorithms in terms of performance
- Multiple coupled perceptrons, in multiple layers → compare with feedforward and convolutional neural networks and similar algorithms
- Apply to the NRR maliciousness classification problem for rogue nodes

Future Work Directions

1. Language Models: can they be applied to 5G networks for security?
    a. Initial investigations with ChatGPT suggest that we can treat the data flowing around in a 5G network as a `language'
    b. Usually messages are transmitted with `headers', `footers', `prefixes' and/or `suffixes' which serve various purposes. For example, Figure 2.55 from (Bertsekas-Gallager, `Data Networks') shows the ATM header format for the user-subnet interface and for the internal subnet. The header contains parts indicating payload type, reserved or non-reserved, priority bit indicator, etc.
    c. AI can be used to study and learn this language and thereafter be usable by the network manager for querying the state of the network - what is the current level of security threat, etc.
2. Consciousness:
    a. Can Baar's `Theatre of Consciousness' model be updated in the light of 5G concepts such as `network slicing'?
    b. What about Integrated Information Theory?
3. MRFs: Apply Berger's Markov Random Fields paper to 5G networks with slices determined by roguishness levels.
4. Ising Model: Explore the application of the Ising model to axon networks and then to rogue-infested 5G networks.

Acknowledgments

The author would like to express his gratitude to Prof. Vireshwar Kumar for encouraging initial work in old directions.

References/Bibliography

1. Gentry, Craig. A fully homomorphic encryption scheme. Stanford university, 2009.
2. Berger, T. (1988). Information theory in random fields. In Coding Theory and Applications: 2nd International Colloquium Cachan-Paris, France, November 24–26, 1986 Proceedings 2 (pp. 1-18). Springer Berlin Heidelberg.
3. Baars, B. J. (1997). In the theatre of consciousness. Global workspace theory, a rigorous scientific theory of consciousness. Journal of consciousness Studies, 4(4), 292-309.
4. Tononi, Giulio, Melanie Boly, Marcello Massimini, and Christof Koch. "Integrated information theory: from consciousness to its physical substrate." Nature Reviews Neuroscience 17, no. 7 (2016): 450-461.
5. Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent. "A neural probabilistic language model." Advances in neural information processing systems 13 (2000).
6. Kline, E., Ravi, S., Cousins, D., & Rv, S. (2022, April). Securing 5G Slices using Homomorphic Encryption. In 2022 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 43-48). IEEE.
7. Mangazeev, V. V., Dudalev, M. Y., Bazhanov, V. V., & Batchelor, M. T. (2010). Scaling and universality in the two-dimensional Ising model with a magnetic field. Physical Review E, 81(6), 060103.
8. McCoy, Barry M., and Tai Tsun Wu. The two-dimensional Ising model. Harvard University Press, 1973.

Bibliography contd.

9. Fine, Terrence L. Algorithms for Designing Feedforward Networks. Springer New York, 1999.
10. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. In Comm. of the ACM, 21:2, pages 120-126, 1978.
11. Chawla, A. (2017). On axon-axon interaction via currents and fields. University of South Florida.
12. Chawla, Aman, and Suma Manjunath. "A Note on Nonlinearities in Ephaptic Axon Tracts." (2023).
13. Bertsekas, Dimitri, and Robert Gallager. Data networks. Athena Scientific, 2021.