Amorphic Encryption

Egger Mielberg

egger.mielberg@gmail.com

27.01.2019

Abstract.
As a symmetric as an asymmetric scheme requires a key (session or private) to be hidden. In this case, an attacker gets a chance and time for finding and decrypting it. As long as a secret has static attributes (length, type of characters, etc.) it will always be vulnerable for an attack.
We propose a new concept of keyless encryption, "*Amorphic scheme*", which is semantically secured and has "Perfect Secrecy" level. It allows a secret to be transmitted over any public channel with no public or private key to be generated and stored.

1. Introduction

Currently there are two main schemes for encryption of data, symmetric and asymmetric.

In case of symmetric scheme, there is only one key, session key (sk), which is used in both ways, encryption and decryption of user information. The same session key must be obtained by both sides, a sender and a recipient. As soon as the sides got the key they can start messaging.

Disadvantages of symmetric scheme:

1. ”No user authentication”.

For example, if Bob wants to transmit some secret data to Alice, he will strongly need to be 100% sure that a message received back from Alice is the original one.

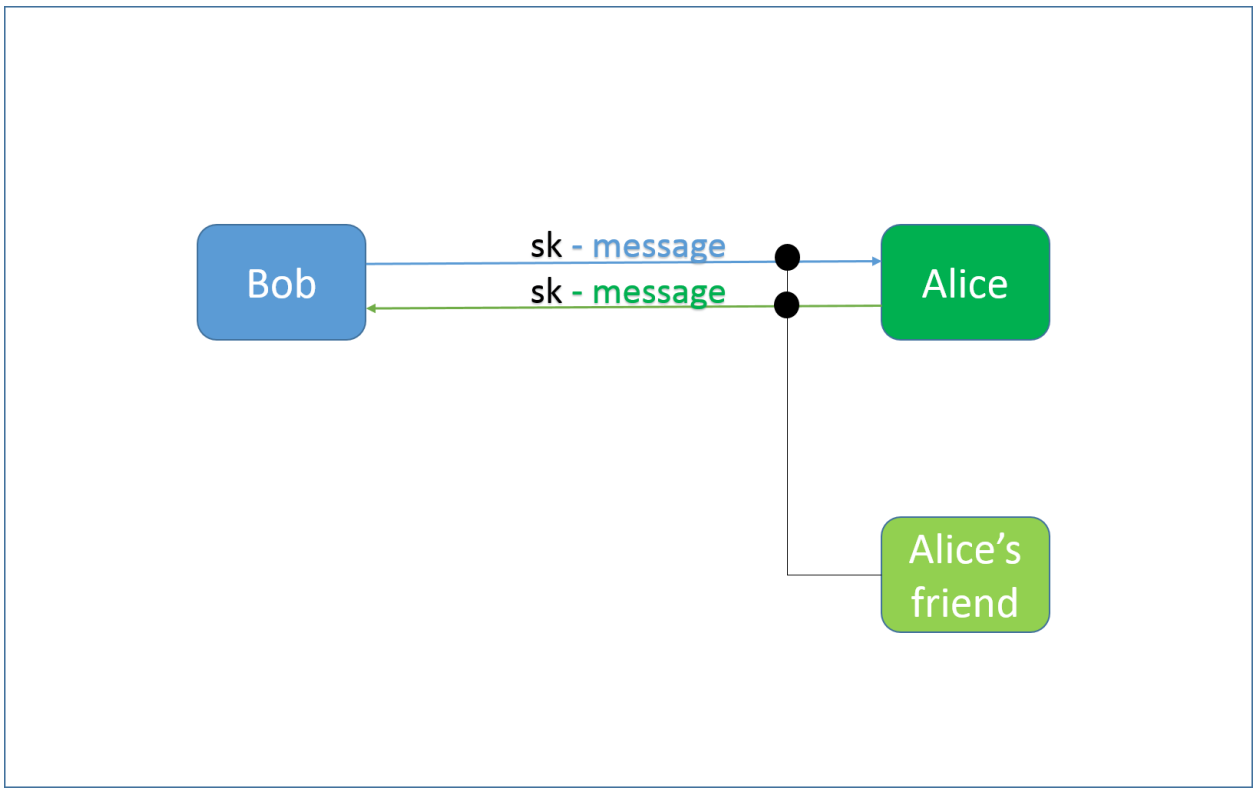
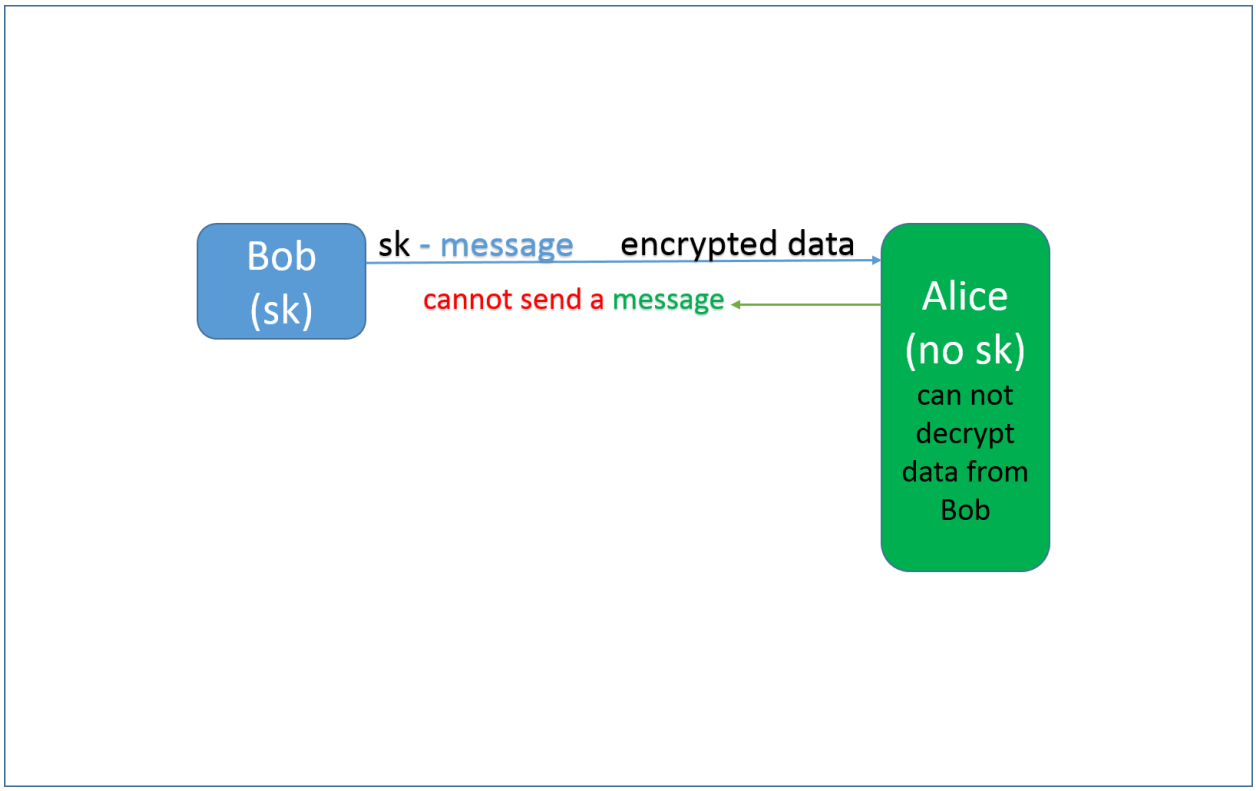


Figure above shows that the symmetric scheme does not propose a mechanism of authentication of a recipient.

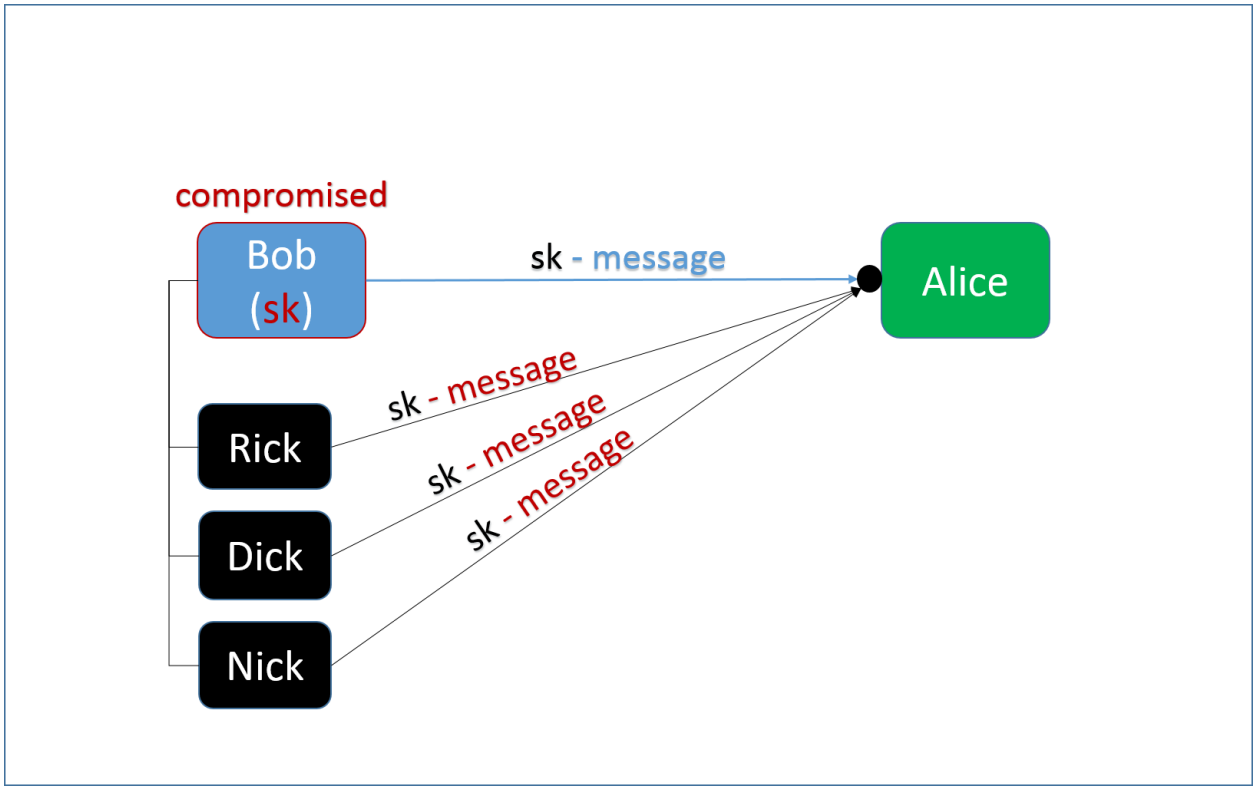
2. ”Simultaneous acquisition of the key”.

In order to start exchanging data both sides, Bob and Alice need to get the same key, simultaneously.



3. ”Key compromise”.

A secure storage place for the key is strongly required. As soon as the key is compromised, anyone will get a chance to masquerade as a sender as a recipient.



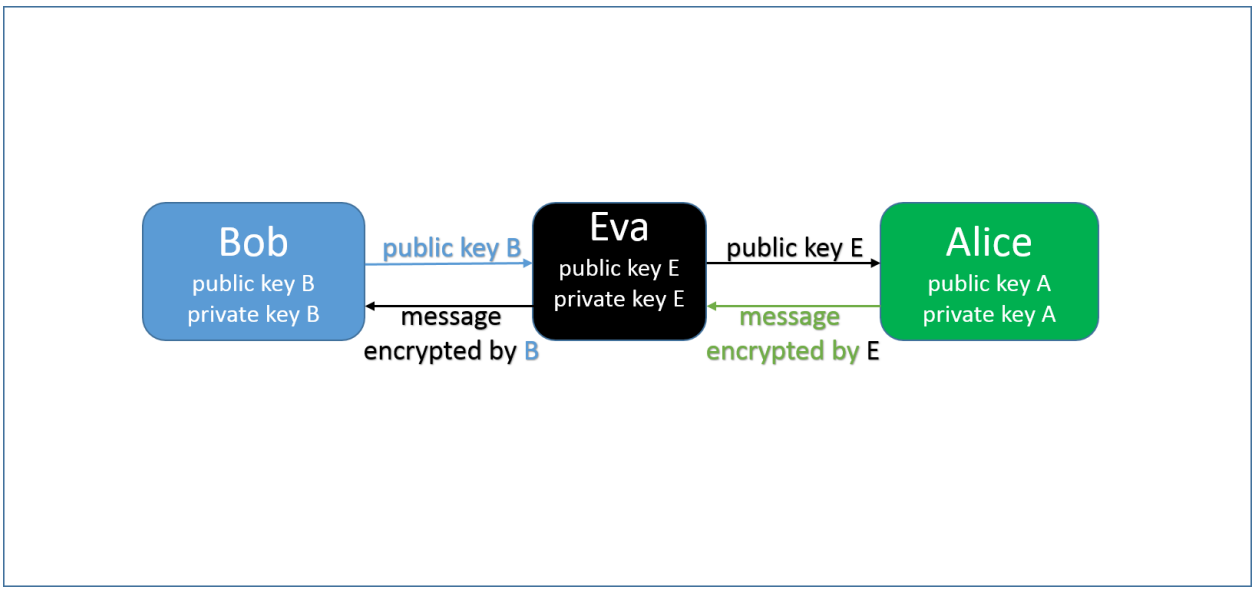
The sides must take an extra care of key storage. Ideally, the key should be duplicated and kept off site in order to protect it against robbery, program bugs, etc.

Asymmetric scheme uses two different keys, public and private. The public key is used for encryption of data. The private key is used for decryption of the data and must be stored in a private offline place.

Disadvantages of asymmetric scheme:

1. “No public key authentication”.

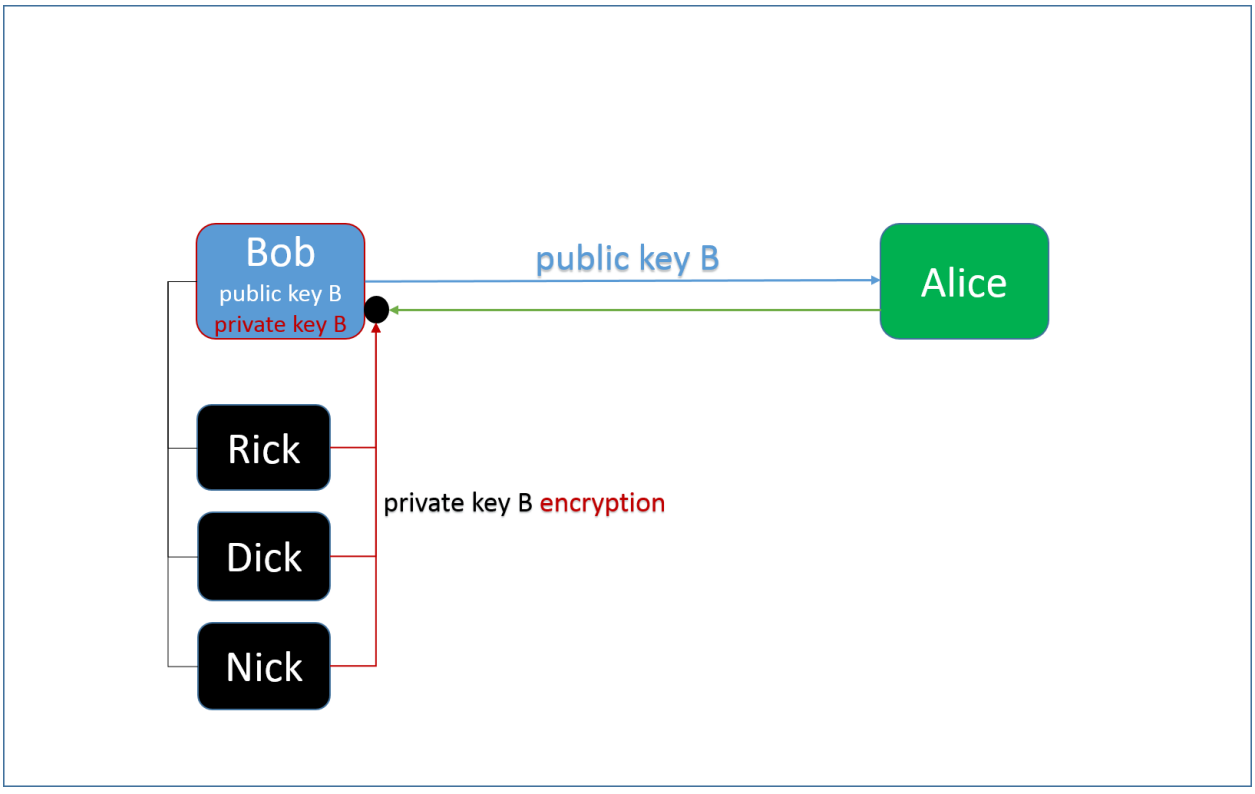
The public key is usually transmitted over a public channel. In case of interception, a third-party can masquerade either a sender or a recipient.



As figure above shows, Bob will never know who he is massaging to.

2. “Private key compromise”.

It is a worst case when the private key is compromised by a third-party. It means that all user data are exposed and can be lost forever.



3. “Quantum computer’s thread”.

As many asymmetric schemes are based on NP-complexity of task solution, a quantum computer which is million times faster than any

conventional computer can become capable of calculating the private key at some reasonable period of time.

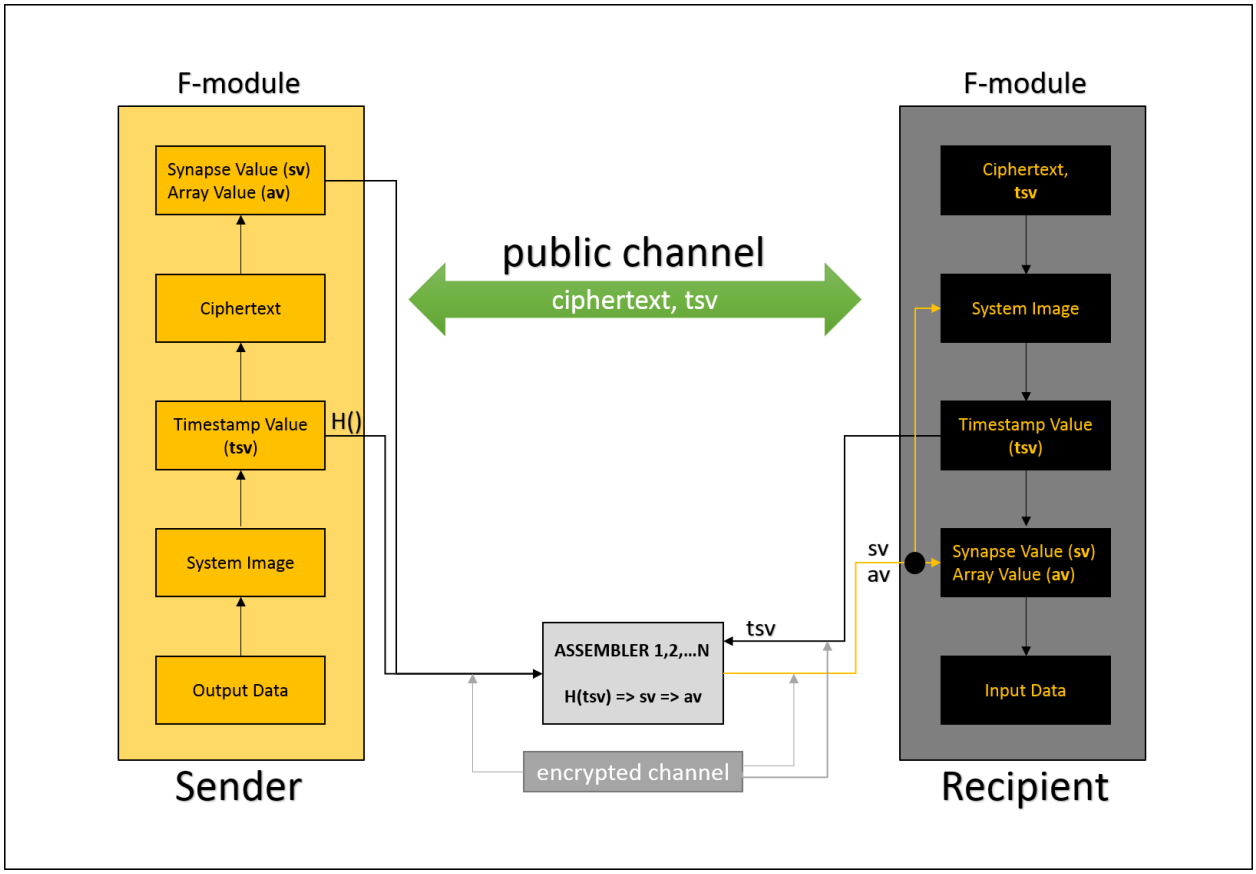
We propose a new scheme, “*Amorphic scheme*”, that eliminates all the above-mentioned disadvantages. It has three main features:

1. No public key is needed for transmitting it over a public channel.
2. No private key is needed for storing it in a private encrypted place.
3. Ciphertext is transmitted in a public channel with no possibility to be deciphered by known algorithms of cryptanalysis.

2. Keyless paradigm

As soon as we create a secret we have immediately to worry about some private and secure place for it. It leads to a creation of isolated and barbed wire protected storage. In other words, we publicly yell where the secret is and state how big the wall surrounding it. In nature, we see absolutely opposite picture. Every piece of information is exposed to a research of any kind.

Keyless concept is about creation of transmission and storage of the secret without isolating it from conventional information.



“Amorphic scheme”

As we can see on figure above, in order to transmit a message to the recipient, the sender needs to pass four steps.

Step 1:

F-module generates a system image. The system image can include but not limited such computer system characteristics as system time, cache value, buffer value, system variable value, etc. Some values in the system can be stored in a specific system (encrypted) file.

This step is crucial for reaching a high level of security and privacy in a process of messaging between participants of Neurochain Network [2].

Step 2:

F-module gets a system time value. This value is needed for:

1. sending it on a public channel with a ciphertext.
2. using it for getting a synapse value from one of the chosen Assembler.

Step 3:

F-module launches a cryptographic algorithm [1]. In many practical cases, it is enough to use not more than 3 rounds. As a result, it generates a ciphertext of amorphic structure.

Step 4:

The cryptographic algorithm forms a synapse and array values. The array value is a part of the ciphertext which is used for deciphering and needed to be found.

The functionality of F-module can be realized by F_{sp} function.

Assembler is a programmable unit which can be as a local or remote module as a full-service server. It stores hash values of tsv and values of sv and av. The location of the assembler can be as permanent as changing on a per communication basis.

The system image plays an important role as for sender as for recipient. The system image is a hardware resource of entropy which includes and strongly tied to *Participant Unique Number* (PUN). PUN as well as system image is generated by a programmable module (application) that realizes *Amorphic Encryption*.

The system image has two main functions:

1. authorization of outcoming messages.
2. authorization of incoming messages.

As figure above shows, while receiving sv and av values, if the recipient is not authorized by the system image for deciphering message from the sender, it will not be granted a right to proceed any further.

High level of amorphous is reached by the following three components:

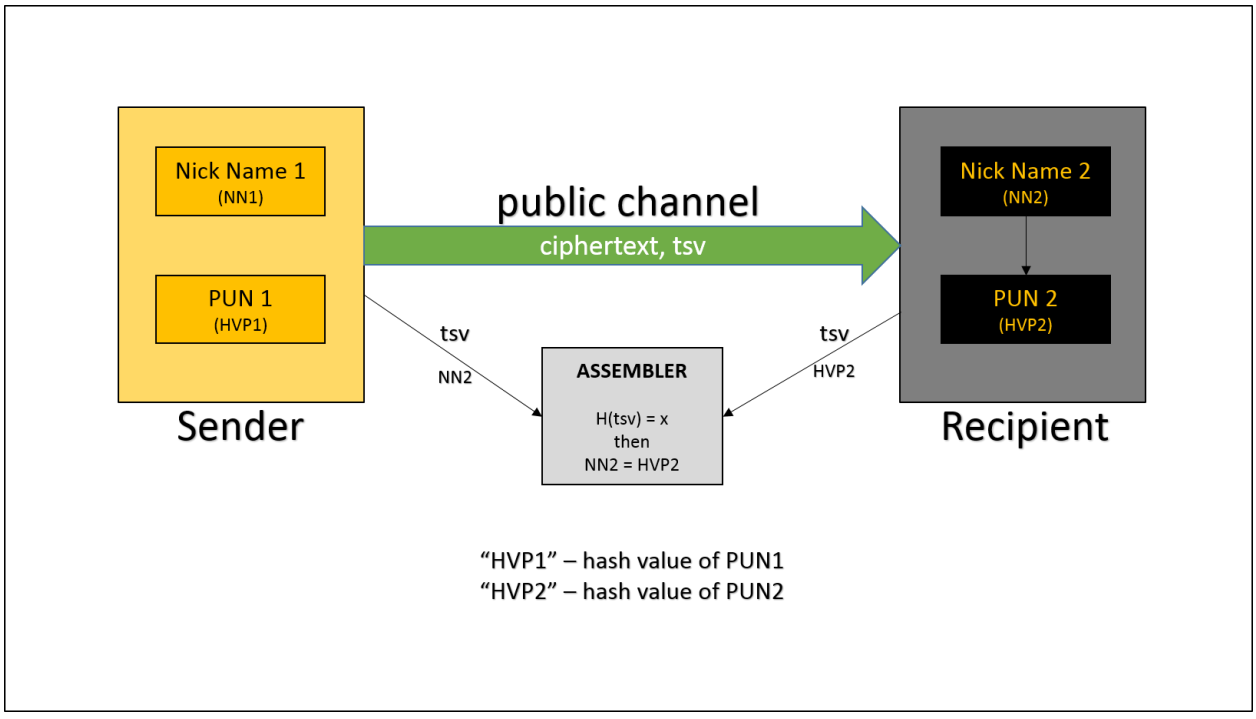
a. system image.
b. timestamp value.
c. av of NACA.

In total, using hashes of the all three components will lead to a strong random value which is dynamically generated from one message to another.

3. Level of secrecy

The level of secrecy of *Amorphic Encryption* can be shown and proved by elimination of the main disadvantages of both symmetric and asymmetric schemes.

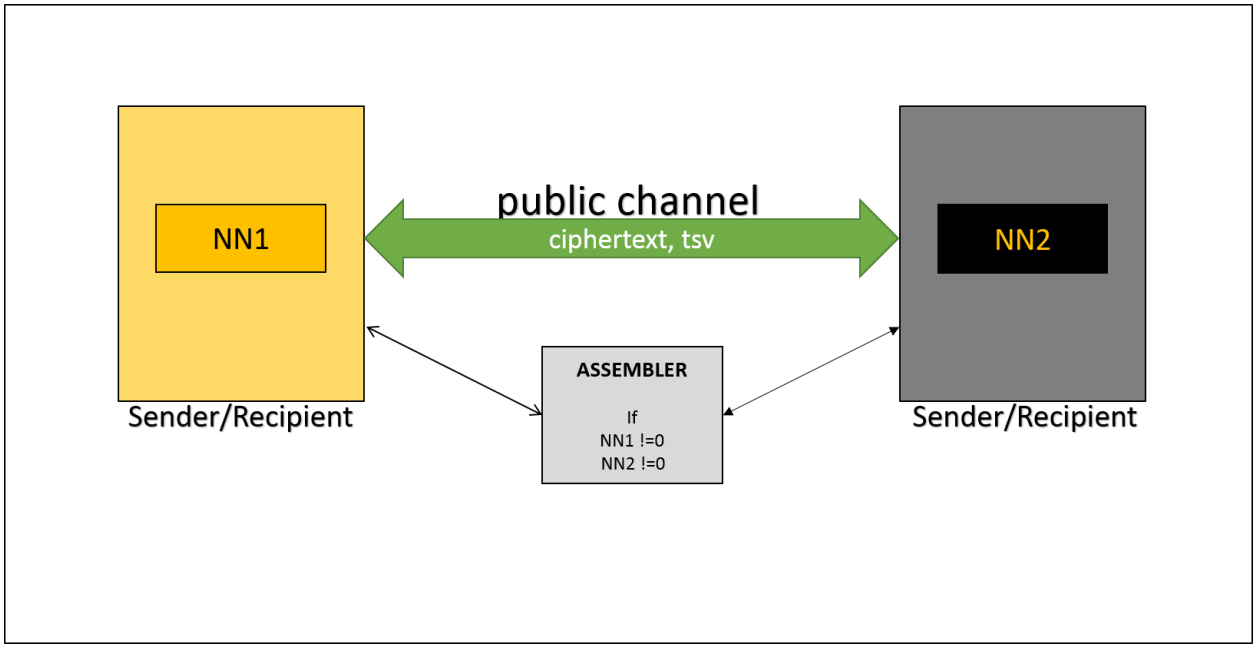
"No user/key authentication".



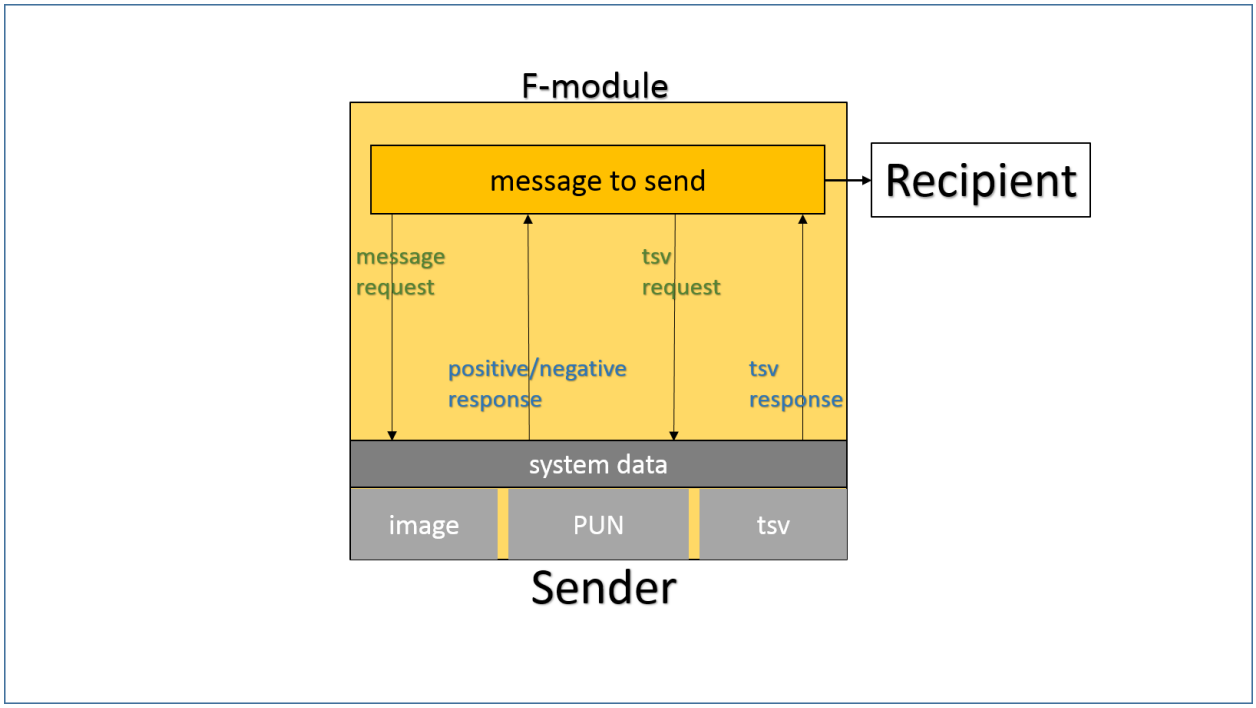
The authentication of participants is realized by a "one-time" tsv check during their first message in Neurochain Network. All subsequent messages between participants are regulated by *Assembler* which are decentralized and use only two parameters, tsv and NN. *Thus, none of the participants needs to transmit neither a public key nor a private key.*

“Key compromise”.

Starting from a second message, a participant will only need to send a ciphertext of the message and tsv. The ciphertext as well as tsv are **always unique**. For example, if a sender is attempting to send the same message for a given period of time, then each attempt will generate a unique ciphertext and unique tsv as well.



There is strict rule of formation of a message. Before any messaging, F-module (submodule) requests PUN from system memory (system file) in order to calculate a hash value of it. If the hash value is matched to the stored one then the message will be sent to a recipient.

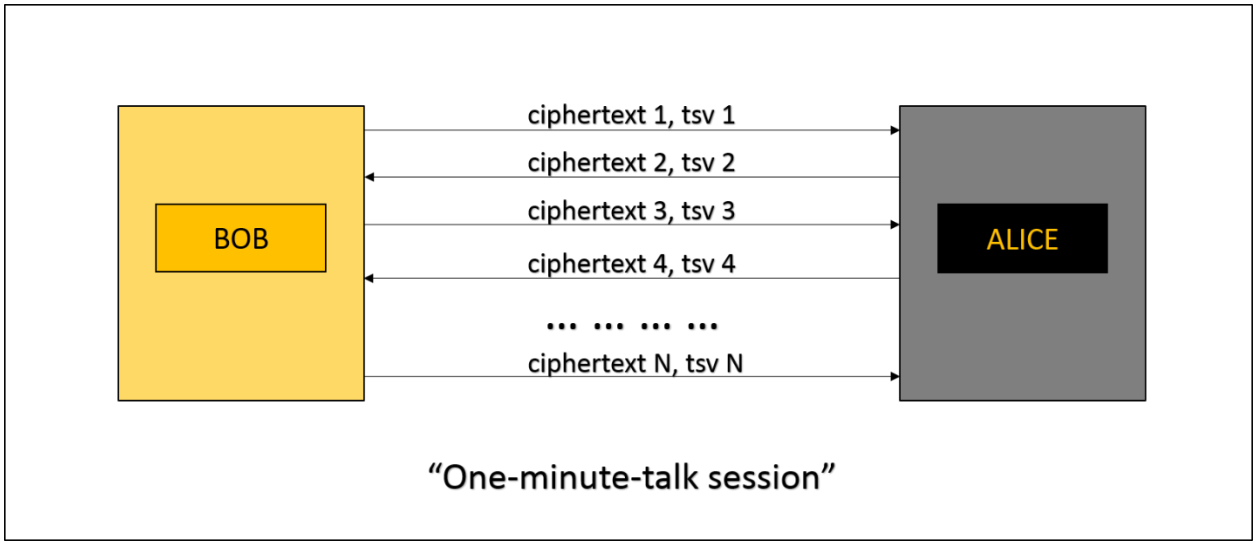


The rule was specifically designed for two security reasons:

1. the authorization process which is realized in a background mode without direct participation of a user of F-module (application).
2. inability to form and send a message by a third-party who intercepted the message or virtualized the user system.

“Quantum computer thread”.

The quantum computers are fit well for tasks with a number of iterative cycles. It is to be specially noted that only a static (constant) data allows to work with itself for some period of time. In case of dynamic data, the possibilities of the quantum computer are significantly reduced.



Neural Entropy [1] of each ciphertext is extremely high and makes quantum computer’s work unsuitable and meaningless.

4. Axioms of Amorphous

An encryption scheme is *amorphous* if and only if the following three axioms are true:

“Axiom of Synapse Value”:

“*There is only one unique synapse value, sv, for a single encryption procedure plaintext-ciphertext*”.

In terms of set theory, we have the following statement:

$\forall m \in M\ \exists E(sv_{\mathrm{i}}, m) = c_{\mathrm{i}} \leftrightarrow \forall m \in M\ \exists E(sv_{\mathrm{j}}, m) = c_{\mathrm{j}}[c_{\mathrm{i}} \neq c_{\mathrm{j}} \wedge i \neq j]$,

where,

$M - set\ of\ plaintexts$,

$sv - XOR(rk, plaintext)$,

$c - set\ of\ ciphertext$,

i,j = {1,2,3,…n}.

"Axiom of Set of Subset":

"*Each round of a single encryption procedure has a unique size of set of subsets (H_i, K_j) of the round ciphertext*".

In terms of set theory, we have the following statement:

$|P(c_i)| \neq \left|P(c_j)\right|$,

where,

$i \neq j$,

$c_{i,j} - set\ of\ i\ or\ j\ round\ ciphertext\ of\ a\ single\ encryption\ procedure$.

"Axiom of Cipher Fragment":

"*The round ciphertext has only one unique cipher fragment, sf, which satisfies the following rule:*

$sf = sv(XOR)H_i, where\ i = \{1,2,3, \dots n\}$".

In terms of set theory, we have the following statement:

$\forall c_i \in C\ \exists a_i[\in c_i \leftrightarrow \forall h_{i-1} \in H\ \left(a_i \in (sv\ XOR\ h_i)\right)]$,

where,

$i = \{1,2,3, \dots n\}$ - number of rounds,

$c_i - set\ of\ i - round\ ciphertext$,

$h_{i-1} - subset\ of\ c_{i-1}$,

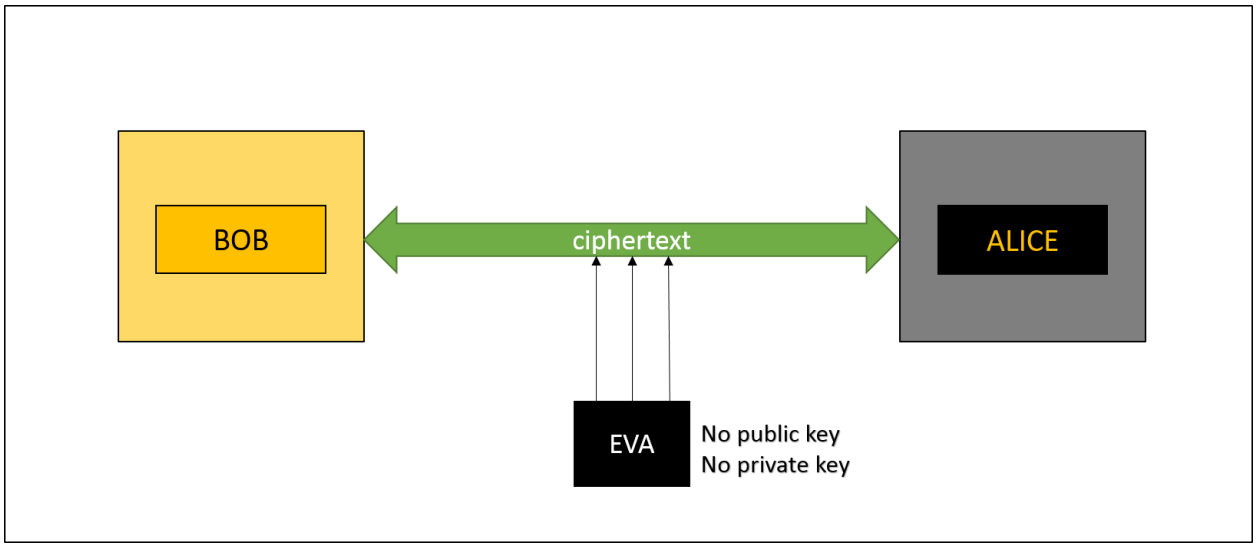
$sv - synapse\ value$ [1].

5. Cryptanalysis

We would like to briefly run on well-known cryptographic attacks in order to clearly show how strong the amorphic encryption concept is.

1. *Ciphertext Only Attack*.

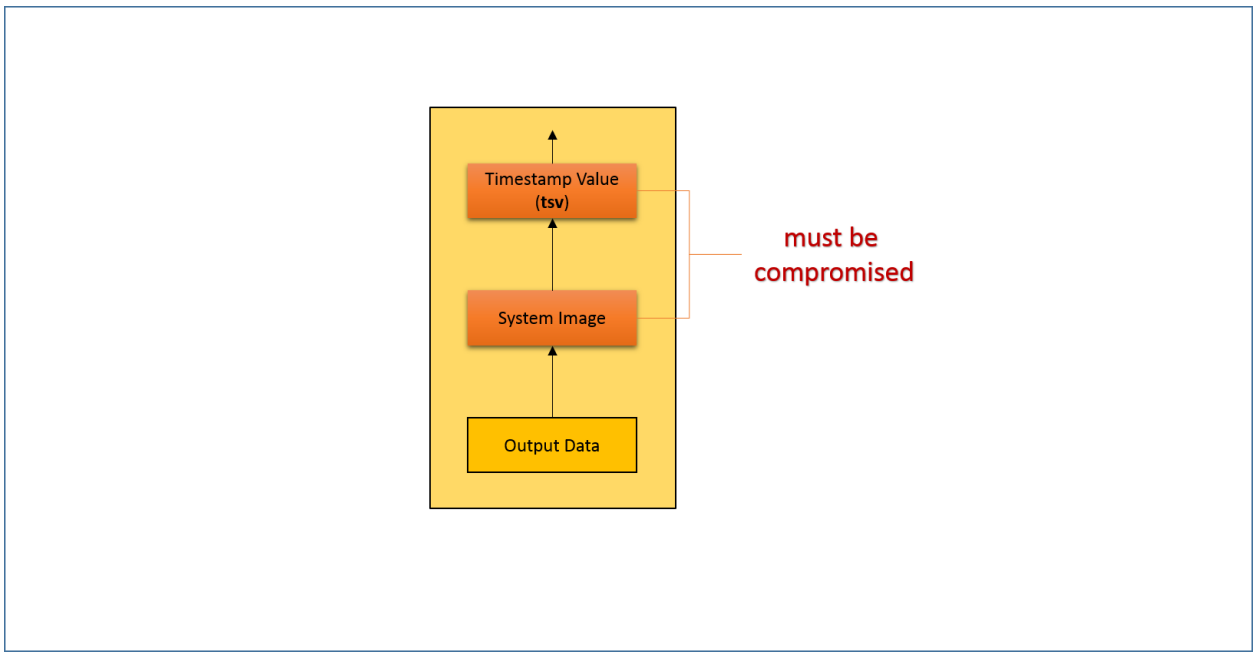
This is a case in which only the encrypted message is available for an attacker.



In this situation, Eva does have nothing but an access to the ciphertext as many other people do in case of *Amorphic Encryption*. Neither public no private key is being created or transmitted. That is why the attacker gets no information about any internal data which was generated inside the system process (F-module) of ciphering.

2. *Chosen Ciphertext Attack*.

This is a case in which a different ciphertext can be chosen with its corresponding plaintext by an attacker.



In the amorphic scheme, there is only one possible way to get known about “key-value” data such as “input-output”. The attacker must break own application’s algorithm for installation of some eavesdropping program. But even if the attacker got the program installed he or she will not get as much information about underlined cryptographic algorithm (NACA) as only a poor probability of resulting ciphertext. There are two reasons for that:

1. system image is highly random value. The probability of getting it with high accuracy converges to zero.
2. There are no two identical ciphertexts for the same plaintext (see *Axiom of Synapse Value*).

In other words, for Bob knowing the main principles of work of his application’s cryptographic algorithm it is not enough to predict the resulting value of Alice’s algorithm.

3. *Known Plaintext Attack*.

This is a case in which both the plaintext and matching ciphertext are available for an attacker.

Actually, in terms of the amorphic scheme, the known plaintext attack is not an attack at all. Each sender of Neurochain Network has an access to what he or she is going to send (plaintext). And all the ciphertexts are public. The high security is taken into account by PUN and system image.

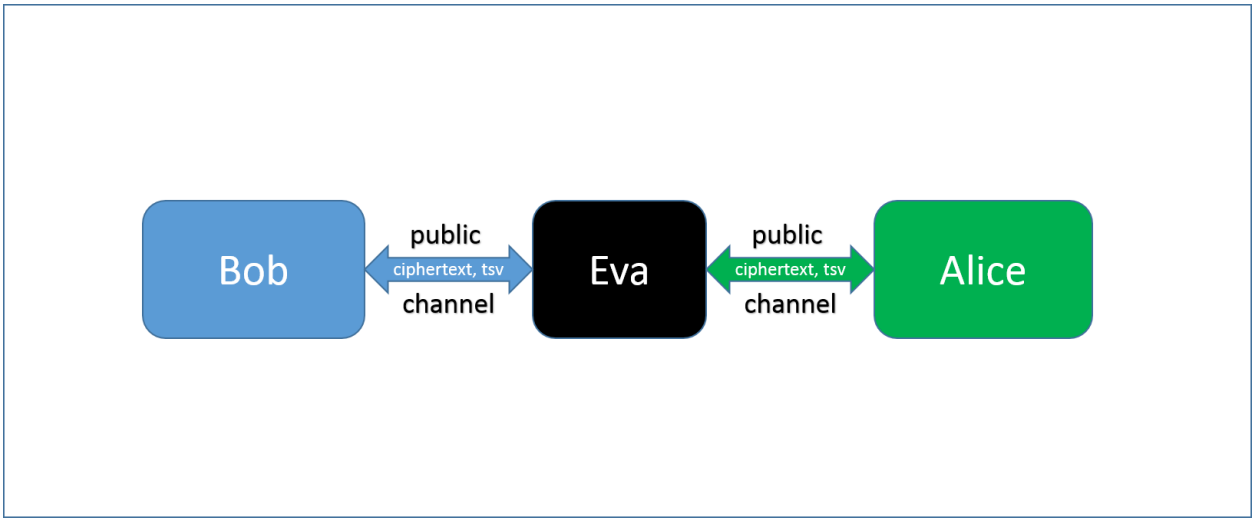
4. *Chosen Plaintext Attack*.

This is a case in which an attacker can choose arbitrary plaintext to be encrypted and then he or she receives the corresponding ciphertext.

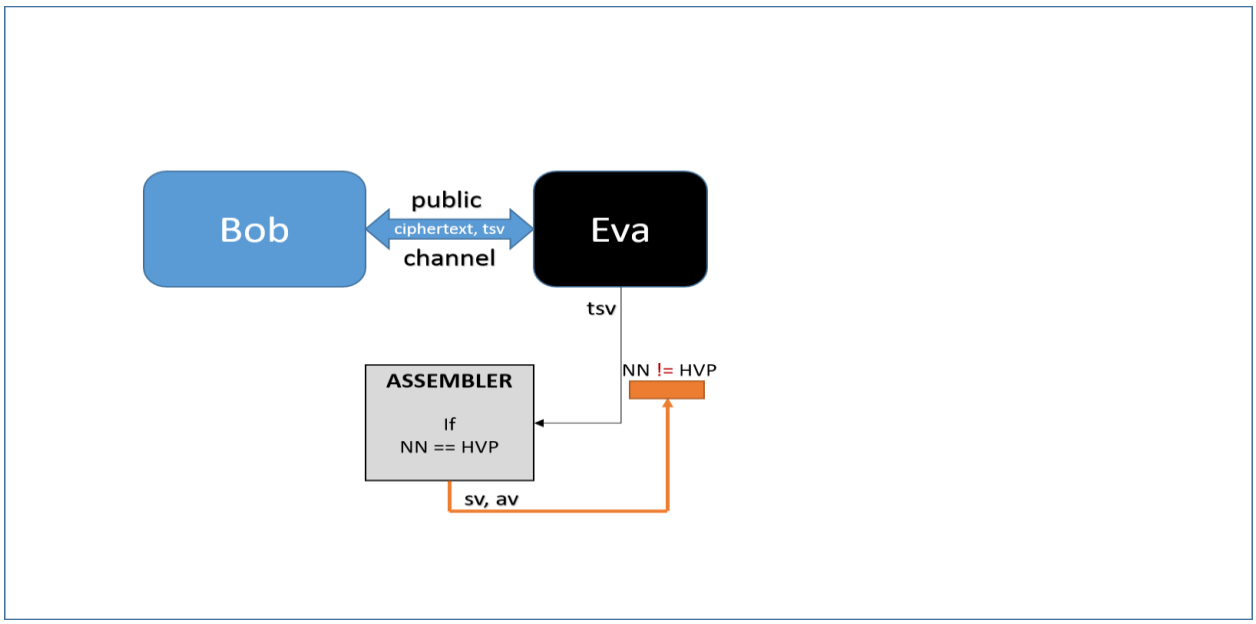
In this situation, there is only one possible way to get some information about underlined cryptographic algorithm. It is a complete break of total victim system. Even after complete break of one of the victim systems, it will not say much about other victim's system and cryptographic algorithm as well (see *Axiom of Set of Subsets*).

5. *Man-in-the-Middle Attack*.

This is a case in which an attacker is able to place himself or herself on the communication channel between two parties.



Even in situation of when Eva is placed herself between Bob and Alice, she will not be able to decipher the message as she will need to get a synapse value and array value, first.



Eva still needs the hash value of PUN of Alice in order to get values of sv and av for deciphering the message. However, a PUN of any participant of Neurochain Network is generated on participant's computer and **never transmitted over any communication channel online.**

6. *Brute Force Attack*.

This is a case in which an attacker tries all possible keys until finding the one that results in the original plaintext. In any cases, the brute force attack becomes a meaningless one because of the following several reasons:

a. there is no private key to be found and decrypted.
b. at least three communication channels (two of which are encrypted) must be intercepted simultaneously ("Alice-Assembler", "Bob-Assembler", "Bob-Alice").
c. at least two over 256 bits hash values must be decrypted simultaneously (HVP1, HVP2).

In case of a per communication basis, the PUN of a participant of Neurochain Network is *dynamically* generated each time when the participant wants to send a message. *In this situation, a super computer or even a quantum computer becomes useless in terms of decryption for a reasonable period of time*.

6. Conclusion

We hope that presented here a new concept of encryption mechanism, Amorphic scheme, will help protect your private data and give you more

thoughts of how to build a robust and new generation cryptographic algorithm of any kind for any purposes.

References

[1] E. Mielberg, “Neuro-Amorphic Construction Algorithm (NACA)”, 2018, https://medium.com/@EggerMielberg/neuro-amorphic-construction-algorithm-naca-f7b563e73288

[2] E. Mielberg, “Decentralized Chain of Transactions”, 2018, https://medium.com/@EggerMielberg/neurochain-decentralized-chain-of-transactions-162a31aee001

[3] M. Huth, “Symmetric Key Cryptography”, https://www.doc.ic.ac.uk/~mrh/430/03.SymmetricKey.ppt.pdf

[4] G. Simmons, “Symmetric and Asymmetric Encryption”, 1979, https://www.princeton.edu/~rblee/ELE572Papers/CSurveys_SymmAsymEncrypt-simmons.pdf

[5] D. Boneh, V. Shoup, “A Graduate Course in Applied Cryptography”, 2016, https://crypto.stanford.edu/~dabo/cryptobook/draft_0_3.pdf

[6] M. Bellare, K. Paterson, P. Rogaway, “Security of Symmetric Encryption against Mass Surveillance”, https://eprint.iacr.org/2014/438.pdf

[7] S. Agrawal, P. Mohassel, P. Mukherjee, P. Rindal, “DiSE: Distributed Symmetric-key Encryption”, https://eprint.iacr.org/2018/727.pdf

[8] D. Pointcheval, “Asymmetric Cryptography and Practical Security”, 2002, https://www.di.ens.fr/david.pointcheval/Documents/Papers/2002_jtit.pdf

[9] S. Yin, L. Teng, J. Liu, “Distributed Searchable Asymmetric Encryption”, 2016, https://www.researchgate.net/publication/312558840_Distributed_Searchable_Asymmetric_Encryption