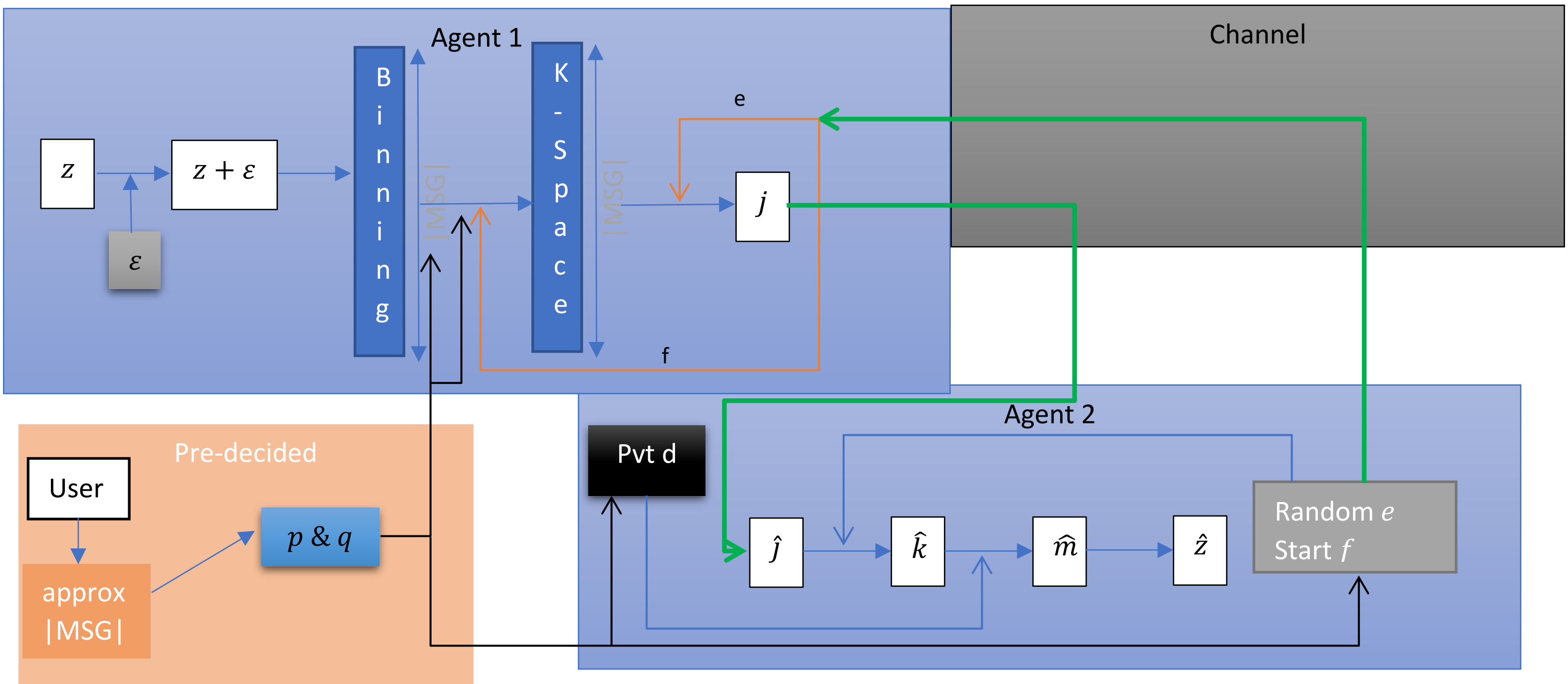
Encryption in MARL Comms

Basic Model

The basic idea mathematical idea behind sending encrypted codes is explained here.



Pre-computations

World starts with, all the agents agreeing on a particular message space size(|MSG|). Pre-computation then predicts mathematically the best available p &q available such that |MSG| of MSG space could be achieved. The calculated |MSG|<=actual |MSG|. Let us assume that p & q calculated |MSG| is same as |MSG| here. This p & q (hence n) is relayed to all the agents before the start of world without channel interfering in it. Each agent then computes series of possible e, and all the equivalence classes (called K-space) here such that |K|=|MSG|.

Active World

Active World starts with each agent choosing a random e, then computing a private _d_ corresponding to e. Also, to deterministically determine the mapping from MSG space to K-space, each agent chooses the initial member f of that K space (i.e 0 maps to f). Suppose Agent 1 wants to send some information to Agent 2 without channel interfering a lot.

So, agent 1 generates a real-valued message Z which can be a vector. An error $\epsilon = unif(-\frac{\delta}{2}, +\frac{\delta}{2})$ is added where δ is the bin width. The resultant vector is binned and then concatenated with epsilon which is then encoded to generate message m. Let φ be some encoding scheme and B be the binning procedure. Thus. $m = \varphi(B(Z+\epsilon), \epsilon)$. This m belongs to the MSG space. Using f & set of e's, Agent 1 determines the mapping of MSG space to K-space. $k = \tau(m)$. This K is then encrypted using the public key value e of Agent 2. Let $j = encrpt(k, e, n)$

Then Agent 1 sends the encrypted message j thorough the channel which is received by Agent 2 as $\hat{j}$. Agent 2 uses it's hidden/private key to decrypt j. $\hat{k} = decrypt(\hat{j}, d, n)$). Agent 2 the uses inverse mapping to get back to MSG space from K space. $\widehat{m} = \tau^{-1}(\hat{k})$. Agent 2 also obtains ϵ from the decrypted message, and obtains $\hat{Z} = \widehat{m} - \epsilon$ which is perceived as approximate Z by the agent 2.

Notations

$\boldsymbol{p}$:	$1st\ Prime\ Number$
$\boldsymbol{q}$:	$2nd\ Prime\ Number$
$\boldsymbol{MSG}$:	$Message\ Space$
$\boldsymbol{K}$:	$K\ or\ Equivalent\ class\ Space$
$\boldsymbol{J}$:	$Encryption\ Space$
$\boldsymbol{n}\|n = p.q$:	$Product\ of\ Primes$
$\boldsymbol{\phi(k)}\|\phi(k) = \{total\ number\ of\ p's\|\ p \nmid k\ and\ p < k\}$:	$Euler\ Toitent\ Function\ of\ k$
$\boldsymbol{\phi_k}$:	$Euler\ Toitent\ function\ value\ of\ variable\ k$
$\boldsymbol{e}\|\ e \equiv\ coprime\ with\ \phi_n$:	$Pseudo - Random\ Number$
$\boldsymbol{d}\|d \equiv e^{-1} mod\ \phi_k$:	$Inverse\ of\ e\ given\ \phi_k$
$\boldsymbol{m}$	$Raw\ Message$
$\boldsymbol{j}\|\ j = k^e\ mod\ n$:	$Encrypted\ Message$
$\boldsymbol{\hat{j}}$	$Recieved\ Encrypted\ Message$
$\boldsymbol{k}\|k = \tau(m)$	$K - space\ \ Message$
$\boldsymbol{\widehat{m}}\|\widehat{m} = \hat{k}^d mod\ n$:	$Decrypted\ Message$
$\boldsymbol{glb}$:	$Greatest\ Lower\ Bound$

Determination of K-space

One of the key features of the algorithm is to figure out what K-space is, or what is the τ mapping. The following lines will explain it.

Let p and q be any two prime numbers. Let

$$n = p.q$$

Let $[E]$ be an array of all numbers e such that e is coprime with ϕ_n.

$$\phi_n = (p-1) \times (q-1)$$
$$[E] = \{\forall\ e\ |e < n\ and\ hcf(e, \phi_n) = 1\}$$

Let there be a msg space MSG such that

$$MSG = \{0,1, \dots, n-1\}$$
$$|MSG| = n$$

For each m in MSG we determine all the mappings using the equation

$$[\tilde{k}] = m^{[E]} mod\ n$$

What we realize is that even tough number of elements in $[E]$ is large, total number of distinct elements in $[\tilde{k}]$ is very small compared to it. That is because it forms an equivalence class i.e The mapping of $m^e mod\ n$ divides the range-space into set of disjoint sets in which each set has the following property:

Let K be the range space and aRb is a mapping such that $b = a^e mod\ n$ for some e in [E]

1) mRm is always present. (If you are considering mappings of some m, same m would always be there in the Range of m)
2) If mRn then nRm also exists. (If m is getting mapped to n for some e in [E], then there also exists a mapping from n to m for some e in [E])
3) If mRn and nRo then mRo also exists. (If m is getting mapped to n, and n is getting mapped to o, then there exists a mapping from m to o for some e in [E])

Let us assume m={1,2,3,4,5,6}. If 2 is getting mapped to {2,3,6} then. 4 is getting mapped to {4,5} and 1 is to {1} then. Let probability of a getting mapped to b is P_{ab} then

Pab	a=1	a=2	3	4	5	6
b=1	1	0	0	0	0	0
b=2	0	0.33	0.33	0	0	0.33
3	0	0.33	0.33	0	0	0.33
4	0	0	0	0.5	0.5	0
5	0	0	0	0.5	0.5	0
6	0	0.33	0.33	0	0	0.33

Thus the equivalence class for n=6 is {{1},{2,3,6},{4,5}}

Then we need to compute the equivalence class size records for n=6 it would be 3. Thus for all possible p,q we compute all possible equivalence class size and store it on ROM. Let K_n be the a maximum equivalence class size of n.

A user picks of some number |MSG|. The algorithm is such that it demines $K_n \sim |MSG|$ such that $K_n < |MSG|$. Once n is found we can use $p\&q$ in our transmissions.

Computation of Public and Private keys.

Agent 2 picks to random large prime numbers.

$$p, q$$
$$n = pq$$

If n is a product of two prime numbers. $\phi(n) = (p-1).(q-1)$ (because of the fact that, prime factorisation of n is pq and for prime numbers $\phi(p) = p - 1$).

$$\phi_n = \phi(n) = (p-1).(q-1)$$

Now, $Agent$ 2 finds out numbers e and d such that

$$e.d \equiv 1 \; mod \; \phi_n$$

Then $Agent$ 2 picks a “Random” number e from $[E]$ such that above equation is satisfied and e is relatively co-prime with $\phi(n)$.

$$e.d \equiv 1 \; mod \; \phi_n \; and$$
$$e \equiv coprime(\phi_n)$$

$Agent$ 2 then calculates d using the euler extended algorithm, $d \equiv e^{-1} mod \; \phi(n)$

Agent 2 then decides a number f which belongs an equivalence class of max size. It computes its τ function as follows.

$$\tau(i) = f^{E[i]} mod \; n \; \forall \; i$$

Each element in range space in τ is unique and if $\tau(i) = \tau(j) \; i > j \; \forall \; j$ then only $\tau(i)$ is considered ignoring the $\tau(j)$

Then the elements in Range space of this mapping is the K-space of f and n

Now information that Agent 2 has, is split into public and private:

$$Public = \{f, e\}$$

$$Private = \{n, \phi_n, d, E\}$$

Encrypting and Decrypting using the keys

First we determine k for the message m:

$$k = \tau(m)$$

Then encrypted message Q becomes:

$$J = k^e \ mod\ n$$

Thus now 2 would be pass to agent 2. Agent 2 uses it private values to decrypt

$$\hat{k} = \hat{J}^d mod\ n$$

We retrieve $\hat{m}$ using inverse relation

$$\hat{m} = \tau^{-1}(\hat{k})$$

Theorem 1

Total number of available e is always increases and is ∞ at $n \rightarrow \infty$

Proof:

Let total number of available e be $avb(e)$

$$avb(e) = \phi(\phi_n)$$
$$glb(e) \approx \frac{\phi_n}{\log\log\phi_n} \approx \frac{\phi_n}{0.834 + 2.3 \times \log_{10}\log_{10}\phi_n}$$
$$\lim_{\phi_n \rightarrow \infty} glb(e) = \infty$$

Thus, which implies as the n increases, ϕ_n increases and hence total number of available e increases. To calculate the exact value of e

$$\phi(\phi_n) = \phi(p-1) \times \phi(q-1) \times \frac{\gcd(p-1, q-1)}{\phi(\gcd(p-1, q-1)}$$

Theorem 2

Theoretical proof of the algorithm

Proof:

For some m;

$$J = m^e mod\ n$$
$$\hat{m} = \hat{J}^d mod\ n$$
$$\hat{m} = m^{ed} mod\ n$$

Remember that

$$ed \equiv 1\ mod\ \phi(n)$$
$$ed - 1 = k\phi(n)$$
$$ed = k\phi(n) + 1$$
$$\hat{m} = m^{k\phi(n)+1} mod\ n$$

$$\widehat{m} = m * (m^{\phi(n)})^k \; mod \; n$$

Remember that $\phi(n)$is a Euler totient function according to which,

$$x^{\phi(n)} \equiv 1 \; mod \; n$$

$$\widehat{m} = m * 1mod \; n$$
$$\widehat{m} = m$$

Theorem 3(Obsolete now)

If channel capacity is limited, we can bend (or wrap) the encryption space into either p or q different subspaces without affecting our ability to decrypt the message. *(Essentially if encryption space is $E = \{0,1,\ldots,n-1\}$ then the new space can be either $E_{small} = \{0,1,\ldots,p-1\}$or $\{0,1,\ldots,q-1\}$)*

Proof:

$$J = m^e mod \; n$$

Let

$$J' = (m^e mod \; n) \; mod \; x$$
$$J' = Kx + K'n + m^e$$

Then

$$\widehat{m} = (\widehat{J'}^d mod \; n) \; mod \; x$$
$$\widehat{m} = (Kx + K'n + m^e)^d mod \; n \; mod \; x$$
$$\widehat{m} = \sum_{\substack{i,j,k=0 \\ i+j+k=d}}^{d} \frac{d!}{i!\,j!\,k!} (Kx)^i (K'n)^j (m^e)^k \; mod \; n \; mod \; x$$

All coefficients having $(K'n)^j$ *where* $j > 0$ will vanish as mod n is there.

$$\widehat{m} = \sum_{i=0}^{d} \binom{d}{i} (Kx)^i (m^e)^{d-i} \; mod \; n \; mod \; x$$

$$\widehat{m} = m^{ed} mod \; n \; mod \; x + Kx \sum_{i=1}^{d} \binom{d}{i} (Kx)^{i-1} (m^e)^{d-i} \} mod \; n \; mod \; x$$

$$\widehat{m} = m \; mod \; x + Kx \left\{ \sum_{i=1}^{d} \binom{d}{i} (Kx)^{i-1} (m^e)^{d-i} \right\} mod \; n \; mod \; x$$

To have $\widehat{m} = m$ we want

1) $m \; mod \; x = m$
2) $Kx\{\sum_{i=1}^{d}\binom{d}{i}(Kx)^{i-1}(m^e)^{d-i}\} mod \; n \; mod \; x = 0$

For 1)

$$m \; mod \; x = m$$
$$\rightarrow m < x$$

For 2)

$$Kx \left\{ \sum_{i=1}^{d} \binom{d}{i} (Kx)^{i-1} (m^e)^{d-i} \right\} mod \; n \; mod \; x == 0$$

$as \; n = p.q$

$$\rightarrow Kx \left\{ \sum_{i=1}^{d} \binom{d}{i} (Kx)^{i-1} (m^e)^{d-i} \right\} mod \; p.q \; mod \; x == 0$$

One of the ways which guarentees that is when

$$\rightarrow x = p \; or \; q$$

$$\rightarrow Kp\left\{\sum_{i=1}^{d}\binom{d}{i}(Kp)^{i-1}(m^e)^{d-i}\right\} mod\ p.q\ mod\ x = 0$$

Final Conditions: $m < x$ & $x = p\ or\ q$;

Hence Proved

Theorem 3

If channel capacity is limited, we can bend (or wrap) the encryption space into either p or q different subspaces without affecting our ability to decrypt the message. *(Essentially if encryption space is* $E = \{0,1, \dots, n-1\}$ *then the new space can be either* $E_{small} = \{0,1, \dots, p-1\}$*or* $\{0,1, \dots, q-1\}$*)*