Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Secret Key Sharing Over Wiretap and State Dependent Wireless Channels

Jibran Ahmed C J, Tonica B S
8th sem CSE,SJCIT,
Chickballapur

Bharathi. M
Associate Professor, SJCIT

***Abstract*— A set of n trusted nodes that aim to create a shared secret key in the presence of an snooper is assumed and using wiretap code to share secret key. Consider a state dependent wireless broadcast channel to remaining nodes including snooper from the trusted node. The trusted nodes communicate over a noiseless and reliable public channel overheard by snooper. The main concept of this paper is to show the sharing of secret key in multiple independent channels by using wiretap coding. The security issues to be improved by state dependent Gaussian channel and using the signal to noise ratio the optimality of this scheme is achieved.**

Keywords*— State dependent wireless broad cast channel, Gaussian broadcast channel, secret key sharing, wiretap code and signal to noise ratio.

I. INTRODUCTION

Assume the condition of generating a secret key *K* among *n* trusted nodes that communicate over a wireless and reliable public channel in the presence of snooper. Restricting the transmissions takes over broadcast channel, where the received nodes are not dependent among receivers of the broadcast transmissions including snooper .The use of state-dependent channels for sharing secret key had gained attention. To show that we can get the same key agreement rate for the entire group as single pair of nodes.Thus the result show that in the presence of reliable, costless and noiseless public channel. After that secret key-agreement for sharing key over the deterministic channel .Using the deterministic channel achievability scheme to get content about the Gaussian wireless broadcast channel with state. To the down utilizing a multi-layer wiretap code based on the broadcast approach of to develop a key-agreement protocol for the noisy and non reliable broadcast problem. Thus it would convert the wireless channel with state to the dependent deterministic case. As a result, show that the secret key generation rate is shared among the nodes in presence of snooper by providing high SNR ratio over state dependent channel that determines the power allocation over different layers of the wiretap code .

II.RELATED WORK

The generation of secret key over wireless channels is a task to be done. Cryptographic analyst named Wyner who coined the concept of wiretap that one can establish secrecy between Alice and Bob by utilizing the noisy broadcast nature of wireless transmissions.However, his scheme works only if had perfect knowledge of snooper channel. And moreover, snooper has a worse channel than Bob. The problem of key agreement between a set of nodes having access to a noisy channel which was visible to snooper which was studied in order of generation of a secret key completely displayed, assuming that snooper does not have access to noisy broadcast transmissions. The state dependent Gaussian channels given by a solving problem can be shown in concept of linear programming. In such concept of linear programming optimization of solution can be achieved. Considering that secret key agreement over wireless broadcast channels with state when snooper also has access to noisy channel transmissions. Thus the scheme for deterministic broadcast channels would be efficient by using specific recursive algorithms.

III. IMPLEMENTATION

Here optimization problem had been introduced and can be solved. Final result is not in closed form but instead propose a recursive algorithm (i.e., a dynamic program) that finds all the possible solutions of KKT conditions (which
provide necessary conditions for an optimal solution to the optimization problem and find an optimum solution by searching among them. By using the proposed algorithm, Thus reducing the complexity of given problem from a multi-dimensional continuous space to a finite elements set; i.e., the set of solutions to the KKT conditions. To better explain our proposed method, the pseudo code of algorithm in Algorithm 1 and Algorithm 2.
Putting it together can describe the algorithm as follows initializes a data structure d(i) for each i for [0 : s] which contains the required information about of Algorithm 1. Then it calls Algorithm 2 that is a recursive function.
Starting from the original KKT conditions, at every iteration, Algorithm 2 picks a number k (such that Ik is not determined yet) and apply Case 1 or Case 2 (depending if it is linear or quadratic case) to that particular Ik.By doing so, the solution of the original KKT conditions (i.e., number of undetermined variables Ik)is reduced by one and may have up to five (in fact up to three if Case 1 holds and up to five if Case 2 holds) new set of KKT conditions to be solved.
Now, it can repeat the above process on each of these new set of conditions and go forward iteratively This procedure is like discovering a tree starting from some point as root (the root is determined by the first k [1 : s□1] picked up by the algorithm). Note that many of these new set of conditions do not lead to valid solutions that satisfy the original KKT conditions This will be determined later as the algorithm proceeds by observing some contradictions on the intervals of Ik's. This process goes until we obtain problems of zero size

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

(that have all variable it is sufficient to check among all of the solutions).

Note that many of these new set of conditions do not lead to valid solutions that satisfy the original KKT conditions This will be determined later as the algorithm proceeds by observing some contradictions on the intervals of Ik's.Thus it use the recursive algorithms to invoke to get optimal solution. Thus many of these new set of conditions do not lead to valid solutions that satisfy the original KKT conditions This will be determined later as the algorithm proceeds by observing some contradictions on the intervals of Ik's. This process goes until we obtain problems of zero content. **Steps of KKT Condition:**

- Algorithm 1 initializes a data structure d(i) for each belongs to [0:s] which contains the required information about Ii*(Initializing the lower bound).
- Then it calls Algorithm 2 that is recursive function.
- Starting from original KKT conditions at every iterations, algorithm 2 picks a number k and apply case 1(linear case) or case 2 (quadratic case) to that particular Ik.

- By doing so , the size of the original KKT condition is reduced by one and may have up to five new set of KKT conditions to be solved.
- Now, repeat the above process on each of these new set of conditions and go forward iteratively. This procedure is like discovering a tree starting from some point as root.
- This process continues until we obtain problems of zero size or at some point in the middle of the algorithm the determined Ik's up to that point violate the KKT conditions.
- The above-mentioned algorithm enables us to find all of the solutions to KKT conditions because the KKT equations provide a necessary condition on the optimal solution, it is enough to check among all of the solutions of KKT equations to find the optimal power allocation for the optimization problem.

Variable	Description
$\mathrm{d}(i)$	An array that contains the available information about the solution I^* of (21) at every step of the algorithm. At the beginning, we have $i \in [0, s]$. However, the size of the problem becomes smaller in every iteration.
$\mathrm{d}(i).\mathrm{l}$ and $\mathrm{d}(i).\mathrm{u}$	Lower and Upper bounds on the indices of states such that we have $I^*_{\mathrm{d}(i).\mathrm{l}} = \cdots = I^*_{\mathrm{d}(i).\mathrm{u}}$.
$\mathrm{d}(i).\mathrm{min}$ and $\mathrm{d}(i).\mathrm{max}$	Determine the interval that I^*_k's belongs to, i.e., $I^*_k \in [\mathrm{d}(i).\mathrm{min}, \mathrm{d}(i).\mathrm{max}]$ where $k \in [\mathrm{d}(i).\mathrm{l} : \mathrm{d}(i).\mathrm{u}]$.
$\mathrm{d}(i).\mathrm{determined}$	Let $k = \mathrm{d}(i).\mathrm{l}$. If the value of I^*_k is completely determined, we have $\mathrm{d}(i).\mathrm{determined}$ = "true" otherwise it is equal to "false".
SolSet	A set that at the end of the algorithm contains all of the solutions (data structures d) that satisfy the KKT conditions (21).

Algorithm 1 Finding all of the solutions that satisfy KKT conditions

Require: s, $\{h_i\}_{i=0}^{s}$, $\{\Delta_k\}_{k=1}^{s}$, $P_{\max}$

1: **for all** $i \in [0 : s]$ **do** ▷ Initialization
2: $\mathrm{d}(i).\mathrm{l} = \mathrm{d}(i).\mathrm{u} = i$ ▷ In general we may have $I^*_{\mathrm{d}(i).\mathrm{l}} = \cdots = I^*_{\mathrm{d}(i).\mathrm{u}}$
3: $\mathrm{d}(i).\mathrm{min} = 0$ ▷ Initializing the lower bound on I^*_i
4: $\mathrm{d}(i).\mathrm{max} = P_{\max}$ ▷ Initializing the upper bound on I^*_i
5: $\mathrm{d}(i).\mathrm{determined} = \mathrm{false}$ ▷ At the beginning, the value of I^*_i is not determined
6: **end for**
7: $\mathrm{d}(0).\mathrm{min} = P_{\max}$; $\mathrm{d}(0).\mathrm{determined} = \mathrm{true}$ ▷ Also part of the initialization
8: $\mathrm{d}(s).\mathrm{max} = 0$; $\mathrm{d}(s).\mathrm{determined} = \mathrm{true}$ ▷ Also part of the initialization
9: SolSet = RECURSION(d, $\{h_i\}_{i=0}^{s}$, $\{\Delta_k\}_{k=1}^{s}$)
10: For all $I \in$ SolSet find the one which maximizes the achievable rate R; call it I**

Algorithm for finding out KKT conditions

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Algorithm 2 The recursive core part of the algorithm that is called by Algorithm 1.

function RECURSION(d,$\{h_i\}_{i=0}^{s}$, $\{\Delta_k\}_{k=1}^{s}$)
  SolutionSet $= \varnothing$
  Find an index j such that $\mathrm{d}(j).\mathrm{determined} = \mathrm{false}$
  if there is such j **then**
    $k = \mathrm{d}(j).\mathrm{l}$
    if $\mathrm{d}(j).\mathrm{l} = \mathrm{d}(j).\mathrm{u}$ **then** ▷ Linear case ($I_k^* \neq I_{k+1}^*$)
      Find the root $r^{(1)}$ of the numerator of $F_k^{(1)}(x)$ defined in (23)
      Based on the value of $r^{(1)}$, break the interval $[\mathrm{d}(j).\mathrm{min}, \mathrm{d}(j).\mathrm{max}]$ if necessary
      for all possible subinterval of the interval $[\mathrm{d}(j).\mathrm{min}, \mathrm{d}(j).\mathrm{max}]$ **do**
        Find the sign of $F_k^{(1)}(x)$ in this subinterval
        According to the sign of $F_k^{(1)}(x)$ (and based on Case 1), update $\mathrm{d}(j).\mathrm{l}$, $\mathrm{d}(j).\mathrm{u}$, $\mathrm{d}(j).\mathrm{min}$, $\mathrm{d}(j).\mathrm{max}$, and $\mathrm{d}(j).\mathrm{determined}$, but to a new data structure d'
        if d' is consistent up to this point **then** ▷ if $\forall i$ we have $\mathrm{d}'(i).\mathrm{min} \leq \mathrm{d}'(i).\mathrm{max}$
          SolutionSet $\leftarrow$ SolutionSet $\cup$ RECURSION(d', $\{h_i\}_{i=0}^{s}$, $\{\Delta_k\}_{k=1}^{s}$)
        end if
      end for
    else ▷ Quadratic case
      Find the roots $r_1^{(2)}$ and $r_2^{(2)}$ of the numerator of $F_k^{(2)}(x)$
      Based on the values of $r_1^{(2)}$ and $r_2^{(2)}$, break the interval $[\mathrm{d}(j).\mathrm{min}, \mathrm{d}(j).\mathrm{max}]$ if necessary
      for all possible subinterval of the interval $[\mathrm{d}(j).\mathrm{min}, \mathrm{d}(j).\mathrm{max}]$ **do**
        Find the sign of $F_k^{(2)}(x)$ in this subinterval
        According to the sign of $F_k^{(2)}(x)$, update $\mathrm{d}(j).\mathrm{l}$, $\mathrm{d}(j).\mathrm{u}$, $\mathrm{d}(j).\mathrm{min}$, $\mathrm{d}(j).\mathrm{max}$, and $\mathrm{d}(j).\mathrm{determined}$, but to a new data structure d'
        if d' is consistent up to this point **then** ▷ if $\forall i$ we have $\mathrm{d}'(i).\mathrm{min} \leq \mathrm{d}'(i).\mathrm{max}$
          SolutionSet $\leftarrow$ SolutionSet $\cup$ RECURSION(d', $\{h_i\}_{i=0}^{s}$, $\{\Delta_k\}_{k=1}^{s}$)
        end if
      end for
    end if
  else ▷ If all $\mathrm{d}(j)$'s are determined
    if the found solution is consistent **then** ▷ if $\forall i$ we have $\mathrm{d}(i).\mathrm{min} \leq \mathrm{d}(i+1).\mathrm{min}$
      SolutionSet $= \{\mathrm{d}\}$
    end if
  end if

Algorithm II called recursively by Algorithm I.

IV DISCUSSIONS AND CONCLUSION

In this sections related to multi party secret key sharing problems and related possible future directions are discussed.
Firstly the SKG(secret key generation) capacity problem among multiple terminals over a state dependant Gaussian channel in presence of passive eavesdropper is still unsolved.
But the optimal solutions of this proposed scheme of SKG has been shown for deterministic channels.
By having intuition from this result the achievability scheme for Gaussian channels is based on message level erasure, simulated by wiretap code.
On the other hand this is still open whether the same connection can be obtained between secret communication over erasure and state dependent Gaussian channels or not.
In this SKG scheme using public channel to send feedbacks from all receivers to senders. But not to transmit all the output feedback.
Hence it is possible to adopt our protocol to use ACK/NAK instead of public channel. However it may not be optimal for deterministic channels. Thus in this work the proposed system used in collaboration with such systems it can add an extra layer of security to the system in physical layer.

V REFERENCES

[1] M. Siavoshani, S. Mishra, S. Diggavi, and C. Fragouli, "Group secret key agreement over state-dependent wireless broadcast channels," in IEEE International Symposium on Information Theory Proceedings, Jul. 2011, pp. 1960–1964.
[2] Y. K. Chia and A. E. Gamal, "Wiretap Channel With Causal State Information," IEEE Transactions on Information Theory, vol. 58, no. 5, pp. 2838–2849, May 2012.
[3] P. Xu, Z. Ding, X. Dai, and K. K. Leung, "A General Framework of Wiretap Channel With Helping Interference and State Information," IEEE Transactions on Information Forensics and Security, vol. 9, no. 2, pp. 182–195, Feb. 2014.
[4] A. Sonee and G. A. Hodtani, "On the Secrecy Rate Region of Multiple- Access Wiretap Channel With Noncausal Side Information," IEEE Transactions on Information Forensics and Security, vol. 10, no. 6, pp. 1151–1166, Jun. 2015.
[5] Y. Liang, L. Lai, H. Poor, and S. Shamai, "The broadcast approach over fading Gaussian wiretap channels," in IEEE Information Theory Workshop, 2009. ITW 2009, Oct. 2009, pp. 1–5.