**Experiment**

The configuration we have used to simulate the proposed privacy-preserving candidate hiring architecture is an Intel i7-12700 processor clocked at a max frequency of 4900MHz and having 12 cores per socket. The simulation was performed sequentially, and the average time was calculated later with the help of sequential time gained from the simulation.

The experiment results have been represented in two different sets in first we compare all 4 models with a encryption key size of 128 bit and later another comparison between the model 2, 3 and 4 has been done with varying the encryption key sizes for homomorphic encryption. Framework 1 has not been considered in second comparison because it is not using any homomorphic encryption, so considering it in second comparison will not be so fruitful as the result for the first framework will be same for all the test cases hence the first framework has been dropped in second comparison.

The experminets have been performed by varying the encryption key size in[128, 256, 512] bits. For 128, all four models have been compared and for 256 and 512-bit encryption model 2,3 and 4 have been compared.

The experiment are performed considering the following two settings after picking an encryption key size from above described list of different key sizes

1. By keeping the number of candidates constant and the number of experts variable.
2. Next, we varied the number of candidates and the number of experts were kept constant.

**Experiment results for setting 1**

The number of experts has varied in [2,4,6,8,10], and the number of candidates has been kept at 20. The result comparing all four models for the encryption of size 128 bit has been shown in Figure 1, and a comparison between the second, third, and fourth models with encryption key sizes of 256 and 512 bits has been shown in figure 2 and 3, respectively.

|  |  |
| --- | --- |
| Framework 1 | Framework 2 |
| Framework 3 | Framework 4 |

Figure 1: Comparing the execution time of all models with an encryption key size of 128-bit.

The results of Figure 1 show that the execution times increase as the complexity of the framework increases. Our framework 1 is very basic model and invovles only digital signature as the step where encryption is used so the computational cost is also very low for the same reason, it took about 0.36 seconds to declare the winner but as the complexity is increasing the the excution time also increaing in later frameworks. If we see the framework 3 is taking about 0.43 seconds for ten number of experts under same settings but the fourth frameworks 0.43 seconds because of the introduction of extra step of encryption/ decryption by the employer it increase the communication overhead as well as computation overhead impaciting overall execution time. Framework 3 and framework 4 are requiring the SMPC to done before declaring the winner hence a more greater communication overhead which is the reason of the extra execution time in these two model in comparison to the execution time of first two frameworks.

The key size also shows a major impact as the encryption key size is increased the execution increases exponentially. In Figure 2 with the encryption key size of 256-bit framework 3 has 0.91 seconds on 10 experts, which is nearly two times what it was for the key size of 128-bit. Similarly, for framework 4 it can be seen that execution time becomes 0.99s for 10 experts as it has one more step then framework 3 as discussed above. When we change the key size to 512 bits the time complexity goes tremendously high which is 3.8s for 10 experts in Framework 3 and for Framework 4 it becomes 4.10s . The same is shown in Figure 3. this trmendous growth is due to the use of homomorphic encryption as described in equation kaha oar hai iski multiplication is oerformed on the ciphertexts to add the original input, but when we are using the 256 bit or 512 bit encryption key size the ciphertext generaed through the HE is equal to key size and perfomring mjltipicain to compute addition of the input on such ciphertext creates a great compuational overhead and it is seen in the result of execution time of framework 3 and 4.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 2: Comparison of the execution time of models 2,3 and 4 with an encryption key size of 256 bits

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 3: Comparison of the execution time of models 2, 3, and 4 with an encryption key size of 512 bits

**Experiment results for setting 2**

For the second setting, when we keep the number of experts constant at five and vary the number of candidates in [10, 20, 30, 40, 50], a similar time difference as of setting one between all four frameworks is found. Figure 4 shows the execution time difference between all four models with encryption key sizes of 128 bits and figure 5 and figure 6 show the execution times of frameworks 2, 3, and 4 for the key size of 256 bits and 512 bits, respectively.

|  |  |
| --- | --- |
| Framework 1 | Framework 2 |
| Framework 3 | Framework 4 |

Fig 4: Comparing the execution time of all models with an encryption key size of 128-bit.

The comparison shows that our framework 1 takes a minimum time which is 0.24s for 50 candidates. Framework 3 and Framework 4 are taking 0.275s and 0.282s respectively for 50 candidates.

Similarly when we start varying the key sizes from 128 bits to 256 and 512 bits huge change is found in frame 2 3 and 4. the reason are already discussed in the previous settings. Execution times of same are shown in figure 5 and figure 6.

one key difference that can be observed from both settings that can be gathered is that execuiton time increases for the higher number of experts in comparison to the number of candidates.

**﻿**The experiment of the fourth framework has also been performed in the same settings as framework 3.

|  |  |  |
| --- | --- | --- |
| **﻿** |  |  |