1. Hello everyone, my name is marko ristic, I’m with the autonomous multisensory systems lab at the otto von Guericke university in Magdeburg, and today I will be presenting our MPI paper: encrypted fast covariance intersection without leaking fusion weights.
2. The context we’re working in considers public networks and distributed sensors, and in particular we’re interested in data fusion over such public networks along with the security implications and guarantees when servers on this network are not trusted.
3. The problem that we consider in particular within this context is the following. We have a natural process being measured by spatially distributed sensors connect to a network. This scenario suits any number of use cases, for example vehicles being localised, or weather being estimated. These independent estimates or measurements are then sent to a cloud fusing server which fuses the data and makes it available for some trusted third party which can query the fuser at any time. The security aspect of the problem is that the cloud fuser is not trusted to learn any information about the estimated state and the third party, although trusted to obtain the fused data, is not trusted to learn individual information from sensors, which may give away sensor locations, accuracies or ranges. The threat model considered is honest but curious, that is, we assume protocols are followed correctly, but any learnt information can be used for malicious gain, which is a common model used in the field.  
   We also note that for a generalisable solution we consider the cross correlations between estimates to be unknown.
4. Before I get into our proposed solution I’ll quickly touch onto some existing methods, the second of which is directly related to our own. Fully homomorphic encryption is an encryption scheme where functions can be evaluated on encryptions without decryption and producing a valid encryption of the result. Although this sounds ideal as a solution to fusion on an untrusted cloud, it is in practice far too computationally expensive to be feasible. Instead, partially homomorphic scheme that only provide part of this functionality are often used. One such method, solves the fusion problem using two such partial schemes and purposely leaks relative sensor errors to allow fusion to be computable at the cloud. The main concern here is that leaked sensor errors can be used by a malicious cloud to infer types of sensors, sensor locations and so on. Our method aims to tackle the problem in a similar manner but without any identifying leakage.
5. So, our method. It’s based on the covariance intersection algorithm and the Paillier partially homomorphic encryption scheme. Covariance intersection is a simple and popular fusion algorithm of estimates and their estimated errors when cross-correlations are unknown and give the final fused result and error with the following equations. On the right the fusion of 2 estimates is depicted graphically where the ellipses show the covariance of estimates and the result covariance of the fusion.  
   While Paillier encryption, as a said before allows some function to be computed on encryptions without requiring decryption. In this case, summation is the only function that can be computed.   
   To explain the method. We’ll first start with the covariance intersection.
6. The idea is to rearrange the covariance intersection into the information form and delay the final parts of computations until the third party queries the cloud. Here we’ve put up what is computed where, but to avoid going into the equations, we in essence say that individual information of estimates is summed (and hence fused) at the cloud homomorphically. The remaining operations, which no longer require information on individual estimate inputs can be computed by the third party after decryption (producing the fast covariance intersection equations, a common efficient approximation of the covariance intersection). From this picture it can be seen that the fusing cloud requires to compute only the summations of information that can be gathered from individual sensors, and this is precisely what’s needed for the Paillier encryption scheme to be useful.
7. The components that need to be summed at the cloud can be produced at the sensors themselves, which in the case of local information filters is already computed with the exception of a few scalar operations. This information can then be encrypted and sent to the cloud, where summations are computed homomorphically. Finally, this combined information can be queried at any time by the trusted third party, which decrypts the sums and computes the final operations which include matrix multiplication and inversion. This naturally adds complexity to the querying party and I’ll go into that in a few slides time.
8. But first, we look at the estimation performance of the proposed method. Since there is no approximation to the weights, the only error introduced is during quantisation of floats for encryption. As it turns out, when key sizes are sufficiently long, that is of a recommended size for good security, these errors are negligible. To the right we can see a comparison of estimation performance between simulations using a normal fast covariance intersection algorithm and our encrypted method.
9. But as I mentioned, although the performance in terms of estimation is unchanged, the complexity of computation is changed drastically to handle encryption and decryption operations as well as a potentially large matrix inversion at the querying party, dependent on state size. For this reason, suitable applications need to naturally consider which objects can be used as an estimator and querying parties as well as state dimension and the number of estimates that will be fused.
10. Now as this is cryptographic security I will quickly mention the security that is met, which is relatively straightforward to demonstrate given the method design. All local state information from sensors is encrypted at the cloud and to eavesdroppers while only total information sum is learnt at the querying party. Since the encryption scheme meets indistinguishability under the chosen plaintext attack it is easy to say that the same is met at the cloud and that the querying party learn only the leaked total sum which gives the final fusion result.   
    Now, I will additional mention that some implicit leakage is always present when elementwise encryption is and an observable network are used. This includes, for example, the state dimension, as this is given by the number of sent encryptions from each sensor and the number of connect sensors themselves. This implicit leakage, however, doesn’t hold any individual information of sensors and it can be said that all identifying information is kept private to sensors.