1. Hi everyone, my name is Marko Ristic, I’m with the Autonomous Multisensor Systems lab at the Otto von Guericke University in Magdeburg, Germany, and today I’ll give a talk on our paper on Privileged Estimate Fusion with Correlated Gaussian keystreams.
2. The broad context of the work is the application and provability of privacy within networks, and more specifically in state estimation within these networks. This has naturally become a greater concern with the increased prevalence of small network devices and the growing accessibility of public networks such as the internet. Problems are typically context specific and can be applied to a variety of networks, including private individuals, commercial parties and government bodies.
3. The concrete privacy problem that we’re interested in is privileged estimation. This problem captures the idea of estimation on a network where measurements are public but some special users are granted a privilege that allows then to get more information out of measurements. The goal in this scenario is that measurements are useful to some degree to everyone on the network but more useful to a select few. An example of this in practice is the original implementation of GPS where a second encrypted channel allowed better position estimation.
4. And as it is security that we’re interested in, in this scenario this is captured by formally guaranteeing a difference in performance between privileged and unprivileged estimators.
5. For those of you present at our virtual talk last year, you may remember a single sensor scheme that achieves these goals. In essence, additive cryptographically pseudorandom noise was added to measurements and removable by those holding a key.
6. Linear systems were considered to accompany cryptographic proofs and resulted in two measurement models holding different amounts of information per measurement dependent on whether an estimator held a key or not. [picture][models]
7. This led to an intuitive difference in estimation performance, where privileged estimators, here green, performed better than unprivileged ones, here red. Ellipses here are the estimate error covariances and are larger for estimates made with more noisy estimates.
8. In addition to the intuitive performance difference, the key aspect of this scheme was the ability to prove the difference in performance in a cryptographic sense. Without going into detail in the presentation, cryptographic algorithms are defined and appropriate definitions including a negligible small covariance are defined before the notion of covariance privilege is presented. This notion allows to capture and prove the performance difference between estimators when models are such that optimal estimators exist, hence the linear models on the previous slide.
9. Now, the scenario we were interested in given this scheme is the presence of multiple sensors. Here, each sensor has a key and estimators can hold some subset of these keys to obtain progressively better estimates than an unprivileged estimator that holds no keys.
10. What makes this scenario more interesting is that now there are two ways in which an estimator can obtain better estimates. Either by removing added pseudorandom noises with a key or by simply fusing more measurements. Of course, when we want to guarantee some privilege gained from holding keys, we want the first case to always make estimates better, but for fusing to only better estimates when corresponding keys are held as well.
11. To formalise the problem similarly to the single sensor case, we consider linear systems to aid cryptographic proofs about differences in estimations, but now have n different measurements. Modification, for now, can be considered an action that produces some modified measurements using a secret key unique to each sensor. A small example on the right shows the measurements of 3 noisey sensors of some path that estimators try to estimate.
12. In addition to the standard linear model assumption we formalise some notation and add assumptions on which sensors can be accessed to simplify notation as well as support number generation which I will get to later. We define an estimator as having access to q measurements and access to a subset p of the measurement keys.
13. The added assumptions we make are on this subset. We say that access to measurements is sequential, that is when q equals 3, an estimator has access to measurements 1,2 and 3, and similarly for privilege, where privilege encompasses having keys up to and including the privilege value p. Lastly, we also assume that when an estimator holds a key it also has access to the measurements associated with that key so simplify computing bounds.
14. This in essence lets us group estimators into three classes. Those with no keys that are unprivileged, those that only have access to the measurements for which they hold keys and those with more measurements than keys, which are all still sequential. It can also be noted here that the only assumption that affects the methods presented is that keys that are held are sequential, while the other assumptions are just there to make notation and explanation easier.
15. Now, with the problem defined, we can describe the performances that we are interested in. That is, how to capture this gain in performance when keys are held or when additional measurements are fused from a cryptographic point of view.
16. The first difference we’re interested in the performance loss lower bound. It is the difference between an estimator that holds keys for its measurements and one that holds no keys but has access to all measurement. Note that if the unprivileged estimator has access to fewer measurements or the privileged one to more measurements for which is does not hold a key, the bound remains a lower bound [explain on pic].
17. The second bound is the performance gain upper bound. It is the difference between the same estimator that hold keys to its measurements and one that has the same number of keys but also access to the remaining measurements. This is an upper bound as access to fewer remaining measurements can only decrease this difference.
18. The goal is for these bounds to be computable and capture minimum decrease in performance possible when not being privileged and the maximum performance increase when fusing unprivileged measurements. And needs to be computable for each privilege p you may be interested in.
19. With the problem and goals set up, we can introduce the method with which they’re achieved. The idea is to correlate the noises that are added at each sensor. This is done by generating uniform samples with each key but rather than creating a Gaussian sample from each one independently, it is done in one go by stacking the samples and performing a single Box-Muller transform with a large correlation matrix. Here, we try to capture the correlated and uncorrelated components with a matrix Z and matrix Y. This result is n correlated Gaussian samples
20. which are added to the measurements from each sensor resulting in extra noise at the sensor when its key is not known.
21. Now what’s important about this generation is that each of the noises now depends on more than one key as they are correlated based on uniform samples generated by different keys. This is important as a privilege p estimator will only hold p keys. A neat trick is that when performing the Box-Muller transform, if a lower-triangular decomposition of the correlation matrix is used, the resulting noises are dependent only on the keys preceding their index.
22. Which, given our assumption about sequential keys being held, means that any sequential subset of keys starting from the first key can be used to reconstruct exactly the associated subset of added noises. The noises from 1 to p, that is.
23. So measurements can be modified and estimators of a certain privilege can remove the noises added by the keys they hold. Since we have a cryptographically sound way of generating pseudorandom noises, these are indistinguishable from truly random noises when the appropriate keys are not held. What this means is that we can compute the exact observable measurement models for the three types of estimators that we introduced earlier.
24. Without going into detail of the equations, some shorthand notation is introduced and measurements from a single timestep are treated as a stacked single measurement. When no keys are held, the measurement model includes additional Gaussian noises added to the existing Gaussian noise and is captured exactly with a greater Gaussian covariance. When all the keys are known, all added noises can be removed and measurements follow the true sensor measurement models. Lastly, when more measurements than keys are available, the dependence of unknown noises on the ones that can be generated has to be considered. The exact measurement model can still be computed and is given by this normal distribution.
25. With known linear models for the three types of estimators we need for the bounds that we compute, we can use optimal estimation to compute the exact lower and upper bound.
26. Recalling the performance loss lower bound, we no know the optimal performance of both of the relevant estimators and when setting a known initial value the result error is the lowest possible for both. Crypotographically we can use the difference between them to guarantee an achievable difference between the two types of estimators for the privilege p.
27. The same approach can be used for the performance gain upper bound again for a privilege p.
28. That sums up the scheme and leaves just a small note on the generation of the correlated Gaussian noises. In the slide previously they are generated centrally.
29. Here a known body with access to all n keys is needed to compute and distribute the pseudorandom noises, but can be done in advance.
30. But another possibility is sequential generation, due to the lower triangular decomposition in the Box Muller transform. Here noises that have been generated so far need to to be passed down the chain so to speak but have no requirement of a central body. The clear downside is that communication grows as n increases.
31. Now all that’s left is to discuss a simulation and look at the results. We consider 4 sensors and here plot the errors over 1000 runs. It’s important to note that the computed bounds earlier are over the average performance of all estimators and that individual runs may not be bound exactly by the bounds. Here, the blue line is the error of an unprivileged estimator with all measurements, the green a privileged one and red a privileged one with additional measurements. As can be expectyed as the privilege increases, the performance gets better drastically, but the additional measurements also allow a slight increase that depends on the estimators privilege. The bounds are here the background colours. The lower bounds lower bounds the difference in blue, while the upper bound upper bounds the difference in red.
32. To explore this further, we can fix the privilege and vary the correlated and uncorrelated components so see how they affect the bounds. As expected, any noise makes unprivileged estimators perform worse, but more uncorrelated noise affects the additional fusion effect of unprivileged measurements.
33. This is still a little hard to interpret and leads us to ask how the two parameters we have chosen affect the bounds that we consider specifically.
34. For this, we consider the traces of the bound series. That is, the average error of the estimators is considered and the difference trace captures the bounds as a single value. Next we consider a stable system and its steady state to see the affect of the parameters on the bounds when they are not changing over time.
35. Now, varying the parameters shows us how the two bounds are affected by the parameters. This is a little dense to understand quickly in the presentation and for those interested the details can of course be found in our paper. In essence it shows that the two parameters are not as independent in affecting the bounds as we would have hoped, but that they do have more of an affect on one bound respectively than the other. Correlated noise affects the lower bound more than uncorrelated noise and vice versa, but the difference in how much they affect the bounds is less meaningful in the uncorrelated noise case.
36. Naturally, this leads to some ideas for future work and extensions. Can we parameterise the correlation matrix in a way that the bounds are affected in a more controllable way? Can we remove the sequential key access assumption required for regeneration of noise by estimators and can we reduce the communication load of decentralised generation of correlated noise?
37. Those are our open questions on this topic and that’s all from me today. Thank you very much for listening and please feel free to ask questions.