Foundations of Cyber Physical Systems

Prof. Soumyajit Dey

Department of Computer Science and Engineering

Indian Institute of Technology – Kharagpur

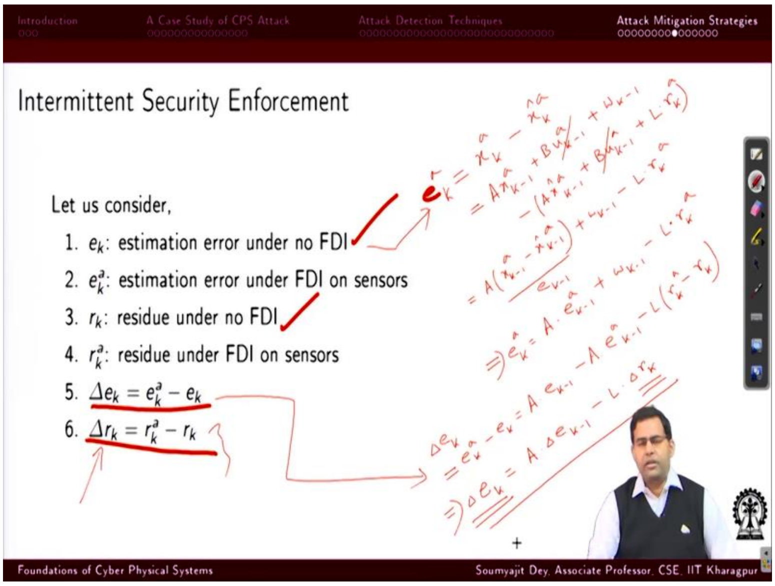
Lecture – 59

Attack Detection and Mitigation in CPS (Continued)

Hello and welcome back to this lecture series on Foundations of Cyber Physical Systems. So, I believe we left it off here where we are talking in week 12 about intermittent security reinforcement. So, I believe what you were discussing had been like Ah well ah we were trying to find out that how sparingly we can use the security primitive inside the CPS so that I mean we do not have to use it always ah leading to bandwidth issue.

But at the same time it is not also fully insecure and is like a just enough secure. So that even under attacks which are still the system does not become unstable.

(Refer Slide Time: 01:05)



Introduction
A Case Study of CPS Attack
Attack Detection Techniques
Attack Mitigation Strategies

Intermittent Security Enforcement

Let us consider,

1. e_k: estimation error under no FDI
2. e^a_k: estimation error under FDI on sensors
3. r_k: residue under no FDI
4. r^a_k: residue under FDI on sensors
5. $\Delta e_k = e^a_k - e_k$
6. $\Delta r_k = r^a_k - r_k$

$$e^a_k = x^a_k - \hat{x}^a_k$$
$$= A x^a_{k-1} + \cancel{B u_{k-1}} + w_{k-1} - (A \hat{x}^a_{k-1} + \cancel{B u_{k-1}} + L \cdot r^a_k)$$
$$= A(\underbrace{x^a_{k-1} - \hat{x}^a_{k-1}}_{e_{k-1}}) + w_{k-1} - L \cdot r^a_k$$
$$\Rightarrow e^a_k = A \cdot e^a_{k-1} + w_{k-1} - L \cdot r^a_k$$
$$\Delta e_k = e^a_k - e_k = A e^a_{k-1} - A e_{k-1} - L(r^a_k - r_k)$$
$$\Rightarrow \Delta e_k = A \cdot \Delta e_{k-1} - L \cdot \Delta r_k$$

Foundations of Cyber Physical Systems

Soumyajit Dey, Associate Professor, CSE, IIT Kharagpur

So, let us continue from where we left off. So, we were We are trying to figure out like well Ah this is our estimation error when there is no FDI. And when there is this false injection ah we kind of wanted to derive the formula which would tell me, what is the estimation error under FDI and in a similar when, what is the residue under no false data injection attack and what is the residue on the sensors, Ah I mean was it what is the residue when there are FDIs happening on the sensors. And then eventually we want to figure out what is the increment in the error and the residue. That means, ah how much extra error is incurred due to the attack in the as part of state estimation. And how much error is in incrementally incurred in the residue due to the attack? So, basically, we want to figure out with and without attack.

What is the ah how bad is the system response? That means how bad is this error and how bad is the residue calculation? That is what we wanted to figure out. Now, if you remember ah in this regard, we have already derived this kind of expressions that well e_k which is it can be expressed like this where W is process noise, observer gain L. These two will get cancelled out. And this is, as you can see the error in the k minus 1 th step in the state estimation.

So, overall we can write this. So, this is like a recurrence equation we are able to obtain on the error with respect to ah the previous error. And the process noise in the previous iteration and these are system property L right the observer gain and what is the residue in the current iteration. So, ah that is your computation of error. Now, this is when this is, there is no FDI happening. Right

And similarly, if we want to figure out what is the estimation error when there is actually FDI happening. So, let us understand what will happen. Then you have this then this is also state under attack. Then this is also state under attack, control under attack right and here similarly you have everything under attack here right, residue under attack. So, what you will get is similar stuff.

So, error under attack Ah error under attack will be related to error under attack in the previous iteration um the noise that is there. The noise is kind of the same right so and the residue under attack. So then what can we see about this? That is the what is the modified error that I am getting? What is the modification or what is the error really induced into the system due to the attack that is happening on some of the sensors?

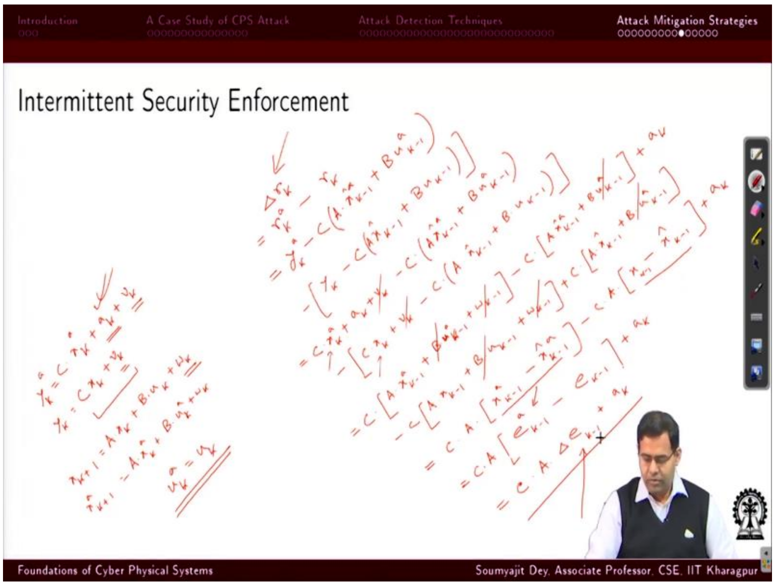
So, we should be able to mathematically write it like this. So, we are basically subtracting error under attack I mean from this we are subtracting the original error. right So, ah what we will have is this minus 1 without attack. So that will give me this minus this and these things will cancel out. right And what I will be left with is here, the residue under attack minus the standard residue. So, I can say minus L times delta r_k, the increment in the residue due to the attack.

So that is how ah the increment in residue due to the attack, will get related with the increment in the state estimation, due to the attack, right fine. ah Let us progress from here and we will also try to come up with a relation for this and then so you can see that what is my end goal? My end goal is to figure out due to the attack. What is the component of state estimation error that is getting injected? Right

So, this is the component in state estimation error and I have been till now, successful in creating a recurrence equation for this component in the state estimation error. But at the same time, in this equation, ah I have this delta r k. right So, what we need to figure out is that well can I have an expression for this delta r k term, using which I can replace it with other system parameters and then I would have ah way to compute this extra error that is this?

So, my end goal is computing this. But now, as I see that I I need to find out an expression for this which I can put in here. And then I can get a more parameterized system property, dependent expression for the extra error that is injected into that attack.

(Refer Slide Time: 08:40)



Introduction | A Case Study of CPS Attack | Attack Detection Techniques | Attack Mitigation Strategies

Intermittent Security Enforcement

Foundations of Cyber Physical Systems | Soumyajit Dey, Associate Professor, CSE, IIT Kharagpur

So, let us try to figure that out. This is what we want to know. Right So, the residue under attack would be the measurement under attack subtracted from what I estimate that state to be. right So, this is the expression for the residue right because in the previous state I have estimated x_k hat using that I can apply it on these and measure what is x_k. This is x_k minus hat from this I can get x_k hat and then I can do this y_k^a minus this. Ah

So that is like the ah error that I have in the residue. So, I am computing ah based on my previous estimate. ah What would be my current estimate of x? And that is x_k hat and then I will multiply it with C you for modelling the output equation y equal to C x and that is kind of giving me that well what is my current expected output. Now, the output that I measure is this and by definition we have the output that I measure minus the output that I have expected Ah to be my residue. Right

So, this is my expression for residue under attack. Now, we also need to find out this right which we already know. It is just the similar expression without the attack. So, so we see that we have y_k^a and y_k here. Now, lets let us try to write them in terms of the states. So, if you recall we had this from our discussion in the previous lecture. This is a measurement noise. This is the attack on the measurement.

And we had y_k when there is no attack then we simply have this is again the measurement noise here. So, if you substitute all those values. So, with this, I can replace y_k^a minus whatever you already had. This is under attack of course. Now, similarly, if you replace y_k with this equation. So, you will end up with this kind of an equation. Right As you can see, these terms will get ah cancelled out here.

And you should be able to write something like this. So, you can see also, here you have the estimates and the controls all in the k minus 1th step and here you have x all in the kth step. right But you are trying to figure out an expression for delta r in the kth step. And typically, we do it like we try to obtain a recurrence from parameters in the k minus 1 th state. We would like to derive the parameter on the k state.

So, we will try to express this using expression for the k minus 1th state. So, if you recall you had these expressions in the process noise. So, let us continue like that. So that is your x_k under attack and the amount of attack in the kth step. So, this is what you can see ah out of this as you can see that many terms would of course get like cancelled out. Right So, you have all the controls ah no well ah yeah sorry, yes. So, this is my expression.

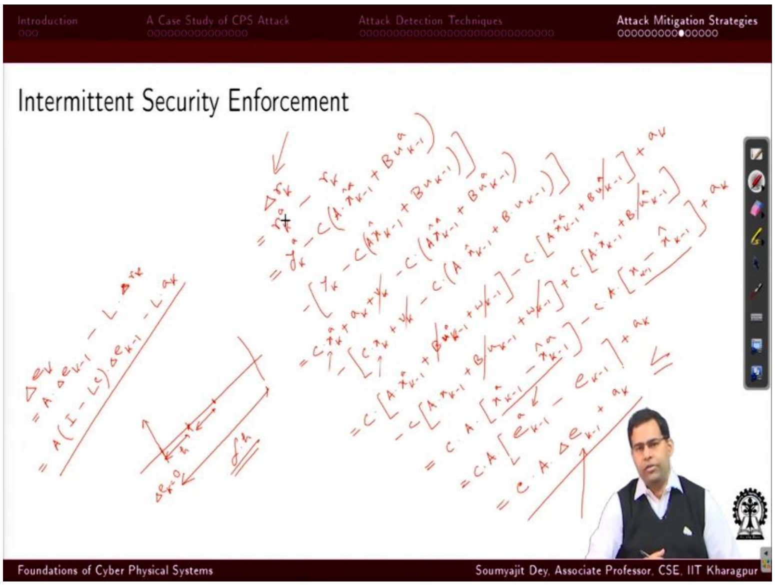
Now, let us observe one thing. What we are doing here is we are trying to figure out the error in the state estimation, under the assumption that the attacks are happening on the sensor measurements only. Right So, it is like, under false data injection on sensor measurement. What is the additional state estimation error that is being injected into the system. That is what we want to figure out. Right

So, in that way, since we are assuming that ah the attack is happening here only and not on the control. So, although in general we write this in an attack scenario. But in this case, since we do not do not have this actuator attack, we have this. right And that would really make these terms cancel out. So, what you will end up with is C times A value of x under attack in the k minus 1 step minus the estimated value of x under attack in the k minus 1th step. right ah

And similarly, you will have this which is C times A, the value of x under no attack. And in the k minus 1th step, and the estimated value of x under no attack in the k minus 1th step. So, this process noise also gets cancelled out right and you have the amount of attack in the kth step. Of course, it is happening only on the sensor side. So, you have C A this is error under attack and so, this is error under attack and this would give you error under no attack scenario.

So that contributes to C times A, the incremental error that you have in the estimate, the estimation error this is the estimation error in both cases and the extra estimation error that is added due to the attack. So that is what you have for the extra amount of residue that is coming due to the ah attack. So, you can see that here you have a nice looking expression we relate which is relating that well the change in residue due to attack with the change in estimation error due to the attack where this is the estimation error under no attack and under attack. Right

(Refer Slide Time: 20:51)



Introduction
A Case Study of CPS Attack
Attack Detection Techniques
Attack Mitigation Strategies

Intermittent Security Enforcement

$$\Delta r_k = r_k^a - r_k$$
$$= y_k^a - C(A\cdot\hat{x}_{k-1}^a + Bu_{k-1}^a) - [y_k - C(A\hat{x}_{k-1} + Bu_{k-1})]$$
$$= C\cdot x_k^a + a_k + v_k - C\cdot(A\hat{x}_{k-1}^a + Bu_{k-1}^a) - [C\cdot x_k + v_k - C\cdot(A\cdot\hat{x}_{k-1} + B\cdot u_{k-1})]$$
$$= C\cdot[A\cdot x_{k-1}^a + Bu_{k-1}^a + w_{k-1}] - C\cdot[A\cdot\hat{x}_{k-1}^a + Bu_{k-1}^a] - C[A x_{k-1} + Bu_{k-1} + w_{k-1}] + C[A\cdot\hat{x}_{k-1} + B u_{k-1}] + a_k$$
$$= C\cdot A\cdot[x_{k-1}^a - \hat{x}_{k-1}^a] - C\cdot A\cdot[x_{k-1} - \hat{x}_{k-1}] + a_k$$
$$= C\cdot A[e_{k-1}^a - e_{k-1}] + a_k$$
$$= C\cdot A\cdot\Delta e_{k-1} + a_k$$

$$\Delta e_k = A\cdot\Delta e_{k-1} - L\cdot\Delta r_k$$
$$= A(I - LC)\cdot\Delta e_{k-1} - L\cdot a_k$$

Foundations of Cyber Physical Systems
Soumyajit Dey, Associate Professor, CSE, IIT Kharagpur

If we go back to our previous expression that we have derived, it was this. The estimation error, the incremental estimation error due to attack in the kth step is related to the incremental estimation error due to attack in the k minus 1th step with the incremental residue using this ah relation that we derive. And now, here you can substitute this value and let us see what we get?

So, of course, this will be substituted here and I would get a nice looking expression. So, you have an A k component also so, minus L A_k. So, what we did is well, we substituted this delta r_k with this. right So then I would have ah A C L coming in in the first term, A goes common, so, I minus L C. right And in the second term I would have L A_k. right So, this gives rise to this dynamical equation on the error in the state estimation.

Now, the question is why are we interested in this quantity. Because this is very much related to how much security I can give. So, you see this is telling me that in one step of control, let us say I have a security reinforcement here due to which this value will be 0 here. But then in the next step, with h as sampling period here the maximum possible value that I can get for the extra ah state estimation error getting injected to the system would be bounded by this value.

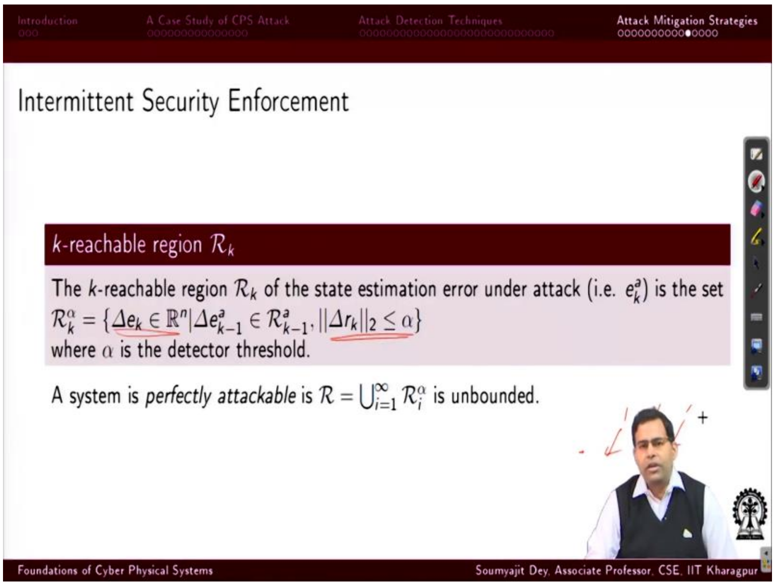
And then again it would be bounded by again this value. right So, if I keep on adding this term over multiple iterations, let us say over some window of length f that means f times h is that on the timeline. So that tells me that well this entire quantity times h or something like that I mean of course I mean not really times h. I mean this is getting into the system and then again the system is evolving and then again another error is coming up like that. Right

So ah that much error ah is getting accumulated into the system due to the attack. right So, if we now, push back this constraint into the dynamical equation, we can actually compute that well how much is the extra state estimation error that is getting into the system, and I can compute this as a function of the number of iterations of the system. Because this is the amount of error that is getting injected extra in one iteration.

So, I will add this up then again evolve the system to the next step. Then again, I will add this much up right ah with changes in k to k plus 1 to k plus 2 like that. And I can compute that in each iteration, how much is getting added up. right In that way, as long as this total error plus the current state will, I mean I mean also the corresponding residue and also in the process ah there is another constraint. That is in every step this residue value must be less than the threshold. Right

This residue value plus the r_k^a, r_k^a that also must be less than the threshold. Right So, under these constraints, the amount of error that is getting added up in the system will be measurable. Once I have this expression that we have created. right So, I can use this expression to create a reach set.

(Refer Slide Time: 25:20)



Introduction
A Case Study of CPS Attack
Attack Detection Techniques
Attack Mitigation Strategies

Intermittent Security Enforcement

k-reachable region $\mathcal{R}_k$

The *k*-reachable region $\mathcal{R}_k$ of the state estimation error under attack (i.e. e_k^a) is the set $\mathcal{R}_k^\alpha = \{\underline{\Delta e_k \in \mathbb{R}^n} | \Delta e_{k-1}^a \in \mathcal{R}_{k-1}^a, \underline{||\Delta r_k||_2 \leq \alpha}\}$ where α is the detector threshold.

A system is *perfectly attackable* is $\mathcal{R} = \bigcup_{i=1}^{\infty} \mathcal{R}_i^\alpha$ is unbounded.

Foundations of Cyber Physical Systems
Soumyajit Dey, Associate Professor, CSE, IIT Kharagpur

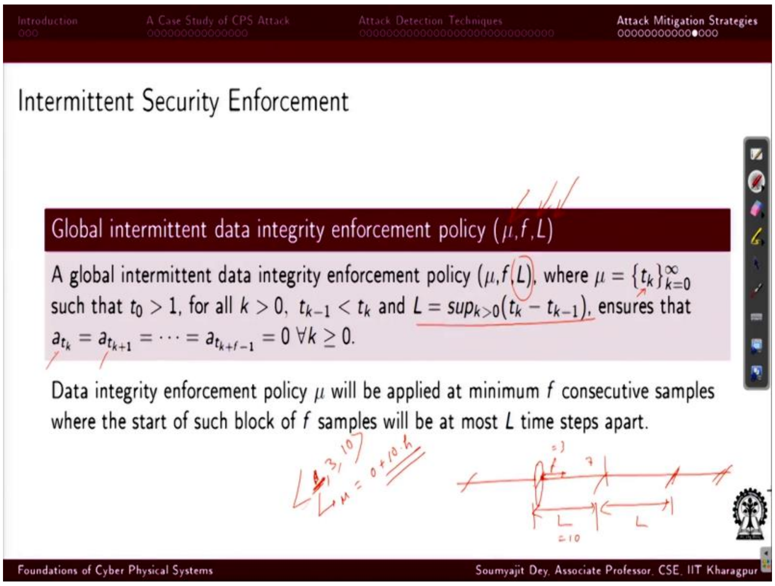
If you remember from our earlier examples that we, I can really compute a reach set of an automaton. Similarly, if I now, create a model of the dynamical system here and I can create what is the reachable region of the state estimation error under the attack ah which is like this reach set. And it tells me ah that will ah as long as my residue ah Is is inside some ah is inside

I mean inside the threshold and the increment in the residue is inside a constraint and in my previous ah iteration I knew what is my state estimation error, I can compute that well what is the extra state estimation error in the current computation. right So, ah based on this ah I can actually see that well ah what is the reachable region in the system under attack with respect to the state estimation error. ok So, as long as these region's, I mean flow flow is within my detector threshold Ah maybe I am ok.

And as long as this reachable region does not lead to any violation of the safety safety trajectories ah of the system the safety boundaries of the system ah I would be ok. ok right And as you can see that if I increase the value of ah this horizon length up to which I am not giving any kind of security enforcement as long as I am increasing that horizon length. ah What is happening is this error value is increasing. right Ah

So then ah this region ah inside which I have this all the possible state estimation error values, this region size is also increase. right ah And eventually it may it may it may cross those safety boundaries which I do not want to happen. right So, even if I choose ah let us say a starting point I know the initial point from which the system starts. And I know that well what is the maximum amount of attack that can happen on each of the you know iterations, I can figure out during the using these expressions that what is the extra error getting getting into the system state estimation and from that extra error I can compute I can get an idea that will ah what is the what is the actual state going to be in which regions in the in the state space. right And then from there I can actually figure out that well ah well I am still inside the safe region or not. So, ah with this, we can go further here.

(Refer Slide Time: 27:58)



Introduction | A Case Study of CPS Attack | Attack Detection Techniques | Attack Mitigation Strategies

Intermittent Security Enforcement

Global intermittent data integrity enforcement policy (μ, f, L)

A global intermittent data integrity enforcement policy (μ, f, L), where $\mu = \{t_k\}_{k=0}^{\infty}$ such that $t_0 > 1$, for all $k > 0$, $t_{k-1} < t_k$ and $L = sup_{k>0}(t_k - t_{k-1})$, ensures that $a_{t_k} = a_{t_{k+1}} = \cdots = a_{t_{k+f-1}} = 0 \; \forall k \geq 0$.

Data integrity enforcement policy μ will be applied at minimum f consecutive samples where the start of such block of f samples will be at most L time steps apart.

Foundations of Cyber Physical Systems | Soumyajit Dey, Associate Professor, CSE, IIT Kharagpur

So, what we can do is we can create a security policy where the policy states that well the policy is parameterized in in terms of these three parameters ah mu, f and L. Mu is nothing but those positions in the timeline where I am going to apply the security ah the security primitive, maybe a crypto primitive or some reduction primitive, something like that. ok So that is ah sequence of those t k values where I apply the security primitive.

And then L is basically telling me that what is the maximum separation this is supremum between these two consecutive values, the differences. All the consecutive time points they are differences, they are supremum. So that means between any two successive security policy application points Ah you take this and this you take this and this. So, what all I am trying to say that these need not be equidistant.

Some of them can be nearby some of them can be larger but their maximum separation is given by L. And f is telling me that well ah after I apply this security policy ah up to ah how much time I have this security policy in place? ok That means say I apply it at t_k right and then I have again this thing at t_{k+1} and then maybe again at t_k plus f minus 1. That means for f length, for f length in the trajectory I have this security policy in place.

And then again, I will start here and this separation is given by L. ok So, inside an L length, I am applying the security policy for f length. ok As I am repeating again, this L can change to some L_1 here but they are, they have a max which is bounded and that is this L value. So, you can You can also simplify this. Let us say I am creating a security policy which is simply saying 1, 3. Ah so So, I mean let us not write mu because it is just a collection of those time points.

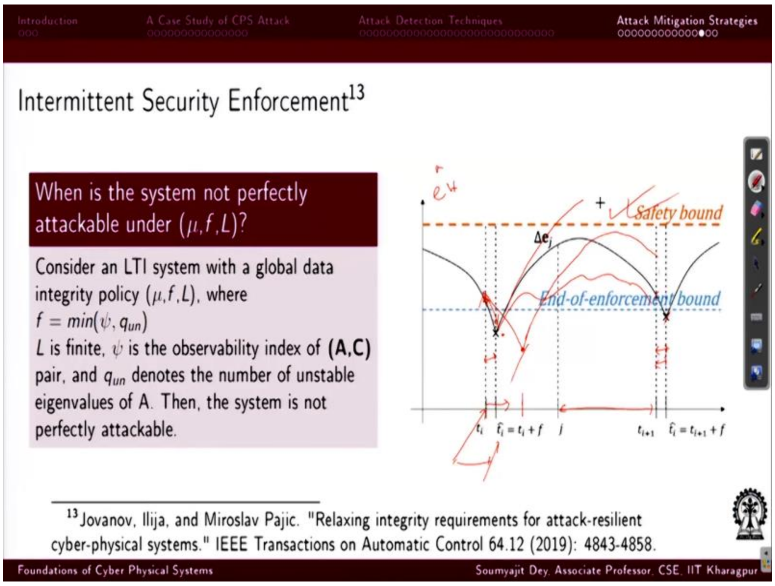
Let us just say that it is 3 we are not writing the mu here we are just writing this keeping it unspecified. We are adding 3 here and we are saying 10 here. That means we are simply saying that well ah let me apply the security primitive in 3 consecutive positions in a 10 length window. That means, after this 3 consecutive position I keep 7 positions where I give no security primitive then again, I just repeat it. Ok

That means my mu value is nothing but it has a 0 offset plus 10 times of h. That means these are the starting times where I start giving the security primitive. And I give it for 3 consecutive iterations. And I I then do not give it for 7 iterations and then again I just repeat. So, this is a simplistic way here I am assuming that the L the separation is all same for all those windows but it need not be same in general.

So, ah again, ah just to summarize, this policy is applied as a me at a minimum f consecutive samples and the start of each block of f samples will be at most be L times separate I mean L time steps apart. So, like this, this is the L time step and this is the L time step here I have f consecutive places where it is being applied and then it is left. So, this is like 3 in our example, the rest is 7. So that L equal to 10 assuming the time steps are all all same.

So, the supremum is equal to all the values. The supremum value is same as all the separation values everywhere.

(Refer Slide Time: 31:34)



Introduction
A Case Study of CPS Attack
Attack Detection Techniques
Attack Mitigation Strategies

Intermittent Security Enforcement[13]

When is the system not perfectly attackable under (μ, f, L)?

Consider an LTI system with a global data integrity policy (μ, f, L), where $f = min(\psi, q_{un})$ L is finite, ψ is the observability index of **(A,C)** pair, and q_{un} denotes the number of unstable eigenvalues of A. Then, the system is not perfectly attackable.

e4
+
Safety bound
Δe_j
End-of-enforcement bound
t_i
$\hat{t}_i = t_i + f$
j
t_{i+1}
$\hat{t}_i = t_{i+1} + f$

[13] Jovanov, Ilija, and Miroslav Pajic. "Relaxing integrity requirements for attack-resilient cyber-physical systems." IEEE Transactions on Automatic Control 64.12 (2019): 4843-4858.

Foundations of Cyber Physical Systems   Soumyajit Dey, Associate Professor, CSE, IIT Kharagpur

So, this this part of the treatment has been taken from ah this paper here. ok So, ah when do I say that the system is not perfectly attackable. So, suppose I have an LTI system where we have this kind of an integrity policy and F is f can be, if F can be created f can be related with the number of Eigen values of the system. Ah And accordingly, we can actually create a condition based on which we can reduce that whether the system is perfectly attackable or not.

But let us not get into that detail. This is just a theorem we wanted to show you we are not interested in the details of the theorem for the part of the course. All we are interested in to is to understand well how I can compute the extra error that gets into the system due to the attack and well once I have one handle on the extra error, how to compute that well with the attack building up over certain iterations how I can characterize the reachable region? And once I am able to characterize this reachable region, well how I can select a security policy? So, let us say here I have an example that we are trying to show that this is the estimation error. That means the entire error that is e_k^a. right We are plotting it as a function of k here. So, this is one point where the security primitive is starting and it is consecutive it is continuing for this t_i plus f amount of time. Ok

And then ah then there is from here to here there is no position where there when there where there is any security primitive applied, these are safety boundary. And the system state estimation error is hovering like this. It could have been something like this. ah It could have been something like this ah etcetera. But what is important is like this that here it must decrease. I have a guarantee that this estimation error will decrease here.

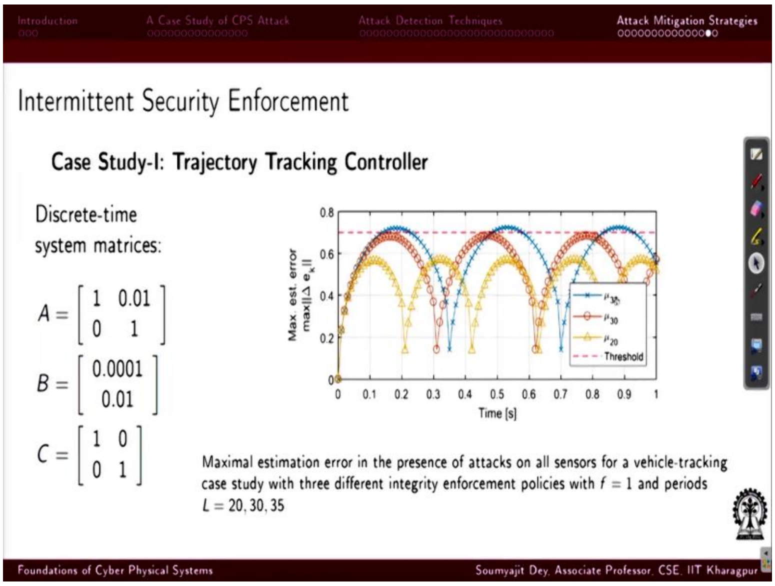
Because I am not allowing any attack here due to the security primitive. Similarly, here I have a guarantee that it will decrease. Because again I am applying the security primitive here and I am making it go below this point. ok The question is how much low I want to make it. I must apply the security primitive up to that number of iterations. So that this unnecessary error that is getting into the system as part of the attack.

But not part of the system noise is reduced that much such that again when I leave the security primitive and let the attack continue note that the attack is also not unbounded. It is bounded by a small value because beyond that it will be detected, attacks are still there. So, I want to reduce it up to that point such that after this, if I leave it on it is own. And the attacks can happen it will not rise beyond this safety boundary. It will be inside this.

So, suppose, for this I see that it goes beyond that means I must secure it for a longer amount of time. So that it should come down somewhere here, so that then if I leave it, it is unable to go beyond. ok So that is the point this curve could have been anything here. I do not have any control on that because of the still the attacks. But I have a control from this point to this point. When I decide that well what should be the length of this interval?

So that inside this interval, I bring down this error to such a point such that from where, if it again rises inside the next interval starting point, when again, the security primitive will come and attacks will be limited but inside this small window it will not be able you do not be able to make the system wear outside the safety bound. So that is the point we are kind of trying to make here.

(Refer Slide Time: 35:21)



Introduction A Case Study of CPS Attack Attack Detection Techniques Attack Mitigation Strategies

Intermittent Security Enforcement

Case Study-I: Trajectory Tracking Controller

Discrete-time system matrices:

$$A = \begin{bmatrix} 1 & 0.01 \\ 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0001 \\ 0.01 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Maximal estimation error in the presence of attacks on all sensors for a vehicle-tracking case study with three different integrity enforcement policies with $f = 1$ and periods $L = 20, 30, 35$

Foundations of Cyber Physical Systems Soumyajit Dey, Associate Professor, CSE, IIT Kharagpur

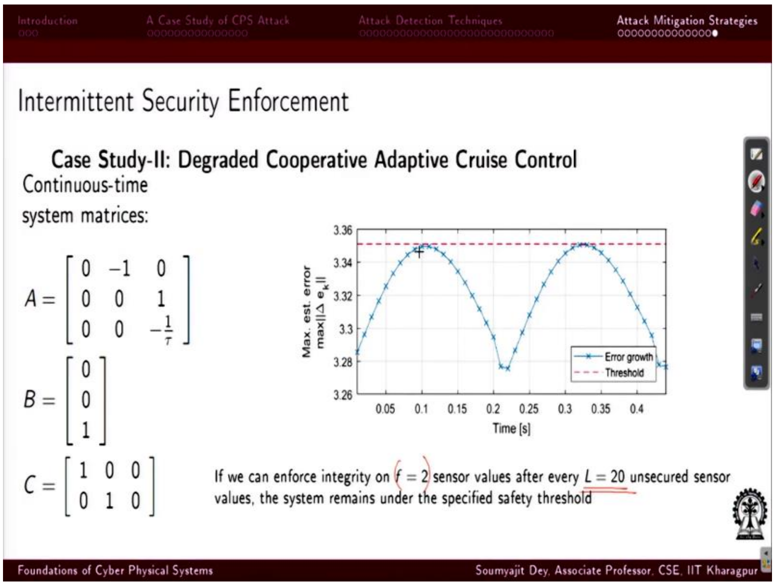
So, ah these authors in this paper, they actually applied this on some trajectory tracking controllers. And as you can see that what they are doing is they are changing the mu value. So, ah let us understand what the mu value means ah it is basically those time steps right when the security policy will apply. So, they are applying it with a period of 20 with a period of 30 with a period of 35.

That means ah at the lower For for this, you are applying it more frequently. You are applying it once every 20 cycles, once every 30 cycles, once every 35 cycles. When you apply it here, your f value is 1, L value is 20, 30 and 35. And accordingly, mu is just 20 times h those time points right mu basically, is the time points when the policy is applied. So, mu is 20 times h, 30 times h, 35 times h, L equal to 20 equal to 30 equal to 35 and f equal to 1 in all these cases. Right

So, when you run it, you can see that here the policy is less aperiodic. So, the error is less the maximum error is like this. In this case, you apply it after every 30 iterations. So, you let the error build up but if you apply it after every 35 iterations you you make the error build up to a larger value. And due to that what is happening is it is crossing the safety boundary. ok So, this is my safety boundary and it is crossing it.

So, I will like to not select ah the security policy, ah to be applied every 35 units. But I would like really like to have it at 30 or better say 20 if the if the system bandwidth allows it to.

(Refer Slide Time: 37:02)



Intermittent Security Enforcement

Case Study-II: Degraded Cooperative Adaptive Cruise Control

Continuous-time system matrices:

$$A = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

If we can enforce integrity on $f = 2$ sensor values after every $L = 20$ unsecured sensor values, the system remains under the specified safety threshold

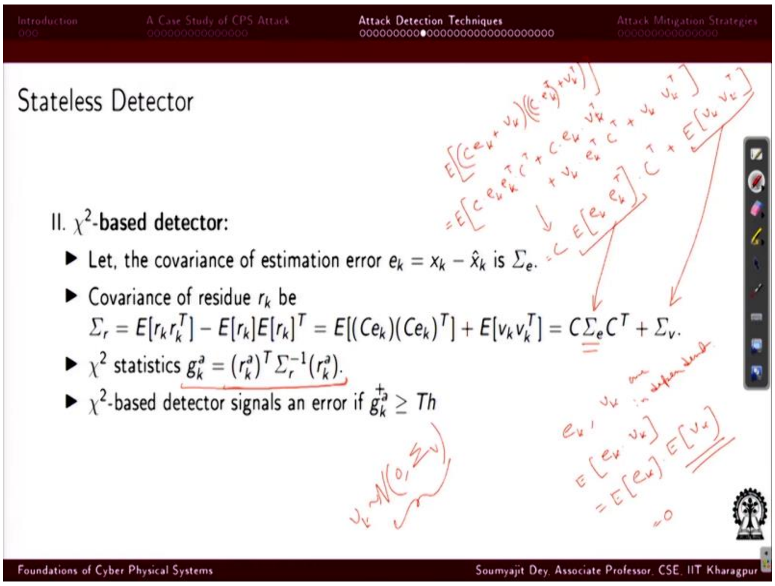
Similarly, you also have an example by these authors from their work on the example of a degraded cooperative adaptive cruise control. So, this was a trajectory tracking system and here you have a cooperative adaptive cruise control. That means you have multiple vehicles and they are following a I mean as minimum separation and they are trying to drive them together. So, this is the system matrix for such a system.

In our tutorials, we will actually show you that well how to build these system matrices for such a platoon, how to build the system matrices for the trajectory tracking system, etcetera. Here we are just starting from the system matrices for the same for the sake of time here. And we are trying to show you that well how these estimation errors change and in this case, we are showing you that how this error is going ah going up.

But the choice of security policy here has been such that well we are able to keep it below this safety threshold. This is the safety threshold here. So, we enforce an integrity on f equal to 2 so, here you see here we are saying that the security policy is such that you enforce the policy once in every 20 iterations. But when you enforce it, you enforce it for two cycles, I mean twice you give the security policy. OK

And so that means, if I give it just once inside 20 then maybe the the the I am not able to bring down the curve on the estimation, error so much. right But if I give it twice ah in inside the ah 20 number of iterations. Then um it stays just below this ah just below this safety line. ok So that is about it here. And also we will like to talk about, ah something else, so, if you remember, we earlier did ah derivation on state, on residue calculation.

(Refer Slide Time: 38:56)


Introduction
A Case Study of CPS Attack
Attack Detection Techniques
Attack Mitigation Strategies
$E[(Ce_k + v_k)(C e_k^T + v_k^T)]$
$= E[C e_k e_k^T C^T + C e_k v_k^T + v_k e_k^T C^T + v_k v_k^T]$
$= C E[e_k e_k^T] C^T + E[v_k v_k^T]$
$v_k \sim \mathcal{N}(0, \Sigma_v)$
e_k, v_k are independent
$E[e_k v_k] = E[e_k] E[v_k] = 0$

Stateless Detector

II. χ^2-**based detector:**

- Let, the covariance of estimation error $e_k = x_k - \hat{x}_k$ is Σ_e.
- Covariance of residue r_k be $\Sigma_r = E[r_k r_k^T] - E[r_k]E[r_k]^T = E[(Ce_k)(Ce_k)^T] + E[v_k v_k^T] = C\Sigma_e C^T + \Sigma_v$.
- χ^2 statistics $g_k^a = (r_k^a)^T \Sigma_r^{-1} (r_k^a)$.
- χ^2-based detector signals an error if $g_k^a \geq Th$

Foundations of Cyber Physical Systems

Soumyajit Dey, Associate Professor, CSE, IIT Kharagpur

So, here we had been talking about this covariance of residue. And we actually try to give you a proof right that well how, for this estimation, for this residue, ah how we can calculate this covariance of residue, and we showed that well why this should be equal to this. But I guess after that we directly wrote this thing. ah So, maybe I would just like to explain a bit on that. So, let us remember what was our expression for residue itself.

Where all these symbols you understand, they have the usual meaning. right This is the error extra estimation error and v_k is your measurement noise. right Now, note that these measurement noise and also the error Ah because we already discussed the error is following a Gaussian here. right So, what will happen is? ah If you take an expectation over this r_k or r_k transpose just by themselves.

ah You are taking an estimation of the mean ah you are you are taking an estimation of this error which is positive or negative with a 0 mean and distributed in a Gaussian manner and v k is also and identically independently, distributed Gaussian with a 0 mean. right So, what will happen is expectation of both this ah will be C times expectation of this which would be 0 C times and plus the expectation of this which would be 0. right Then what remains is expectation of r_k and r_k transpose. Right

So, ah if you just try to compute that ah let us see how that happens. So, here you have r_k transpose. That means this is the transpose over this that would mean e k transpose times C transpose so that is it. So now, if you carry this expectation inside, you will get this from the first term get transpose. And you will have expectation of e_k, v_k and expectation of v_k, e_k like that Okay transpose like that.

Now, so that means I mean these will be the terms that will be 0. And the question is why they are 0. Of course, because we will be assuming that e_k and v_k are independent. So, whenever you will have terms like e of e_k and v_k which will you will get here and here. right You can write them as E of this and E of this and here this will become 0. right So that is why this is with 0.

And you have essentially then C times covariance of the state estimation error which would be given by this sigma e and C transpose then. And then you have the covariance of the process noise. Right So, for the same variable so that is a variance of the process noise, so that is sigma v which is a parameter of the Gaussian distribution. So, you will be assuming that v_k is distributed following ah Gaussian with 0 mean and variance sigma v.

So, this is essentially sigma v and similarly and this is the variance of the error. So, let that be sigma e. So, in this case, what really happens is when you measure the residue well you take it. right And then ah you generate the chi square stat using this expression which is property of chi square stat. And here you are using sigma r which is nothing but the variance of the residue. So, essentially your weight you are measuring r_k under attack.

And you are weighing it with the variance of the residue or basically, the variance of the variable r_k. Right And you calculate the variance using this expression and you apply this assumption that e_k and v_k are independent. Because anyway, we there is a good assumption because e_k is the property of the system and v_k is a property of the environment. right ah So, ah based on that you can come to this expression.

And from here you just apply Ah you just replace this with the variance of error. We are assuming that there is known to you. And the variance of ah the noise which you also consider as a known parameter as part of a noise model. right So, given that we know that the variance of error and variance of noise, ah we can write what is the covariance of residue. And then you use the covariance of residue right here. Ok

So, I believe this express this explanation was missing but now we have it there, so that is pretty much about our discussion. Ah On ah techniques for attack detection techniques for attack mitigation some of these are at a bit advanced level. ah So, we are trying to give you an basic idea that why these are important and how some of these techniques can be applied. Some examples on this we will try to cover in the tutorials.

That we have as part of the weekly I mean weekly tutorials that will have in the course. And I hope with that you will have a fair idea about how ah system properties can be used to secure CPS systems. ok With that we will end our lecture. Thanks for your attention.