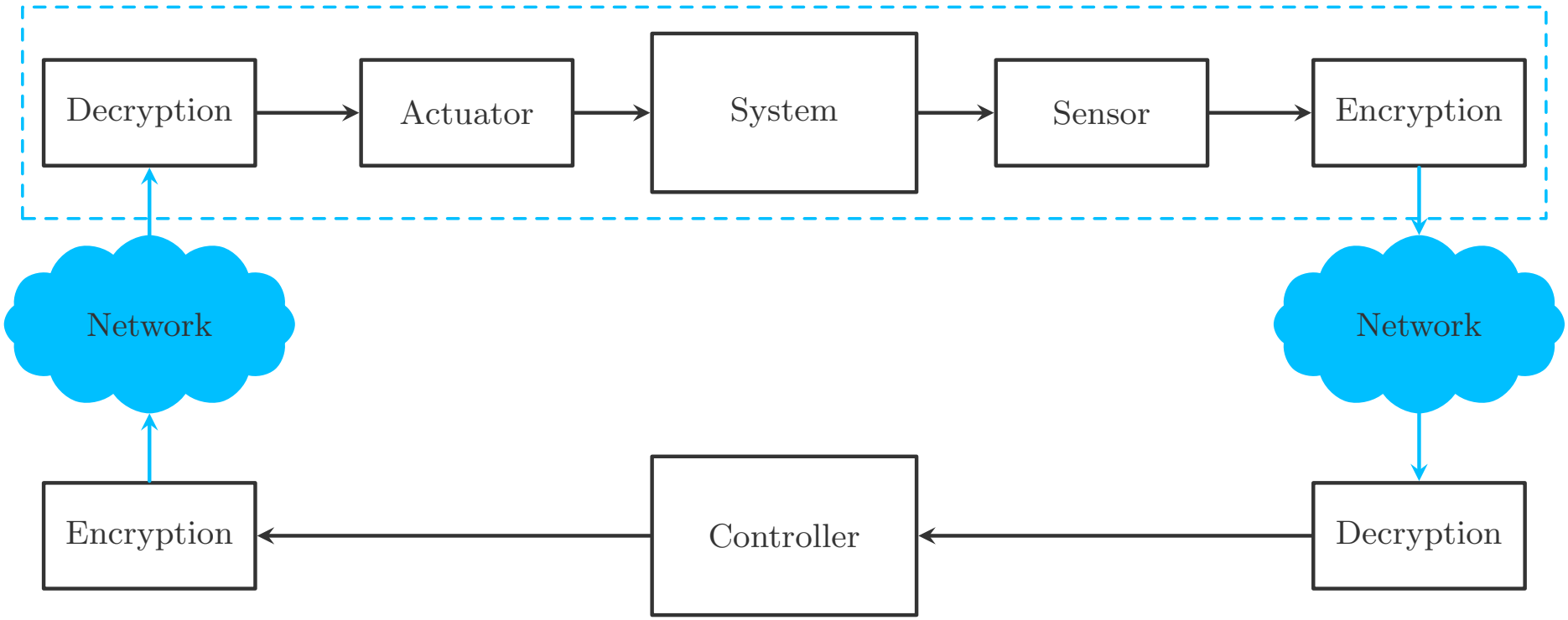
Digital Implementation of Homomorphically Encrypted Feedback Control for Cyber-Physical Systems

J. Tran, F. Farokhi, M. Cantoni, I. Shames

MIDAS LAB
(Melbourne Information, Decision, and Autonomous Systems Lab)
University of Melbourne

THE UNIVERSITY OF MELBOURNE

A (Somehow) Familiar Problem



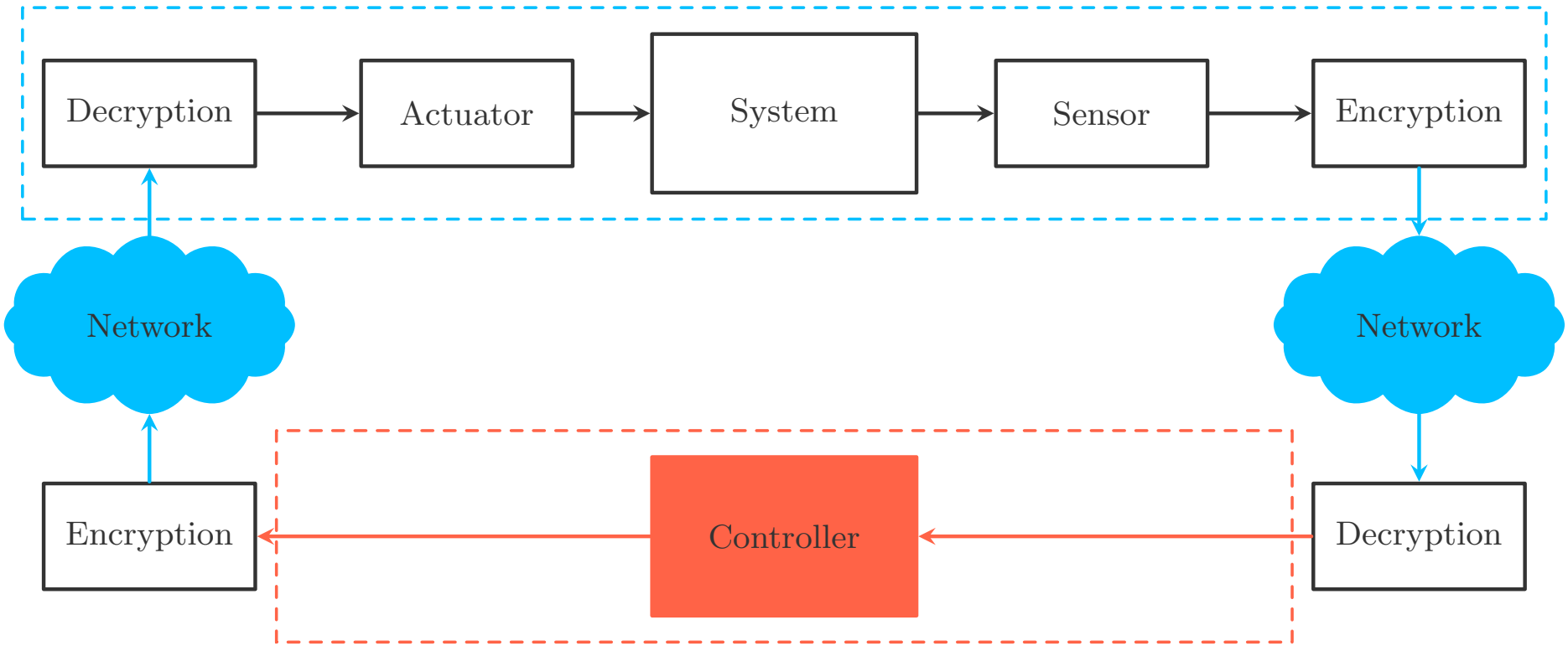
A classical networked control system (NCS) over secure networks:

- A cyber-attacker cannot access network data for
  - Constructing the model of the system;
  - Driving the states of the system to an unsafe state.

A (Somehow) Familiar Problem with One Glaring Shortcoming

"Just because you're paranoid doesn't mean they aren't after you."

– Catch-22



The cyber-attacker can hack the control centre and access all information that s/he needs or the cloud provider is dodgy.

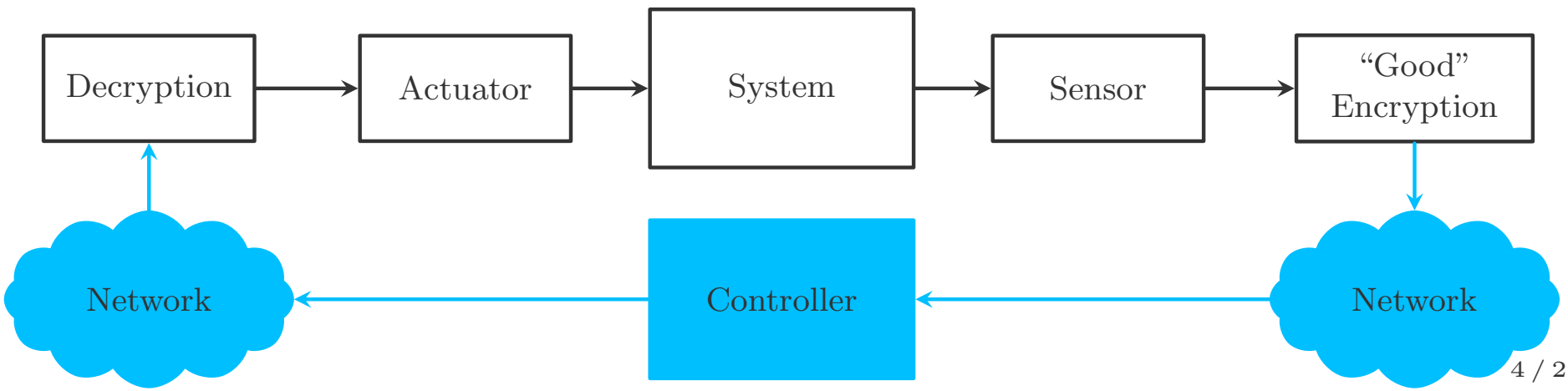
In the pursuit of a solution...

In the proposed solution all external system-related signals must be encrypted while the performance (stability) of the closed-loop is not compromised. The computations need to be completed in a 'timely' fashion.

Travis agree:

"...There's no obvious solution to this plight

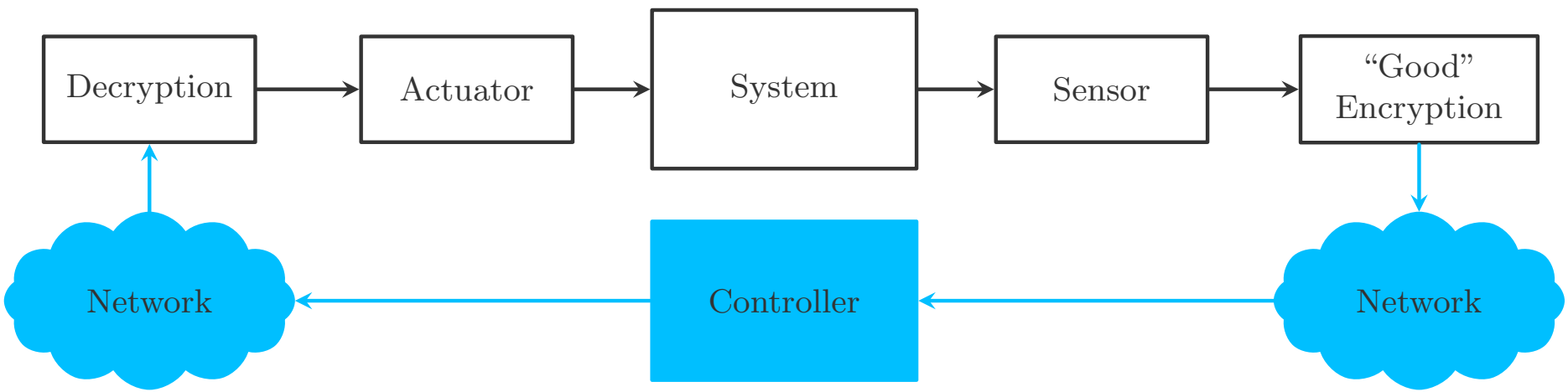
Keep it locked, out of sight"



In the pursuit of a solution...

In the proposed solution all external system-related signals must be encrypted while the performance (stability) of the closed-loop is not compromised. The computations need to be completed in a 'timely' fashion.

Sun Tzu Agrees:
"Be extremely subtle, even to the point of formlessness. Be extremely mysterious, even to the point of soundlessness. Thereby you can be the director of the opponent's fate."



Outline

Outline

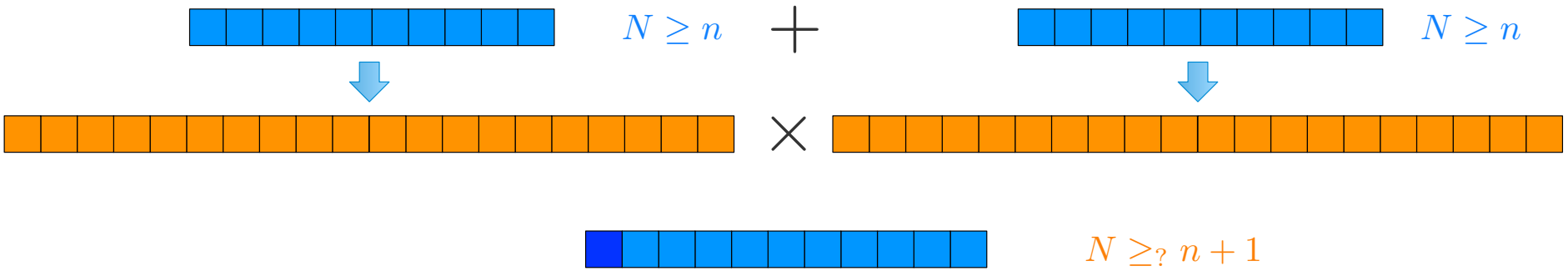
Semi-homomorphic Encryption: Implementing the Pallier Method

"Ford!" he said, "there's an infinite number of monkeys outside who want to talk to us about this script for Hamlet they've worked out."

– Douglas Adams, The Hitchhiker's Guide to the Galaxy

- A semi-homomorphic encryption scheme comes with public key κ_P, private key κ_S, and a group operator $\circ$.
- In Pallier the group operator is modulo multiplication:
  - $\mathsf{D}\left(\mathsf{E}(a, \kappa_P) \circ \mathsf{E}(b, \kappa_P), \kappa_S\right) = a + b$ (ciphertext + ciphertext)
  - a and b are integers.
  - Encryption and Decryption require exponentation and multiplication of large numbers.
  - Large random numbers need to be generated.
  - $\circ$ operator the align is modulo multiplication.

One simply does not add and multiply with impunity...



- Note the extra bit to prevent possible overflow.
- Multiplication (plaintext $\times$ ciphertext) is just multiple additions (ciphertext + ciphertext +...+ ciphertext).
- One should be very careful when it comes to implementing recursive algorithms and dynamical controllers. It is easy to run out of memory.
- Anyhow, matrix-vector multiplication is possible:

$$\text{plaintext matrix} \times \text{ciphertext vector}$$

Outline

```
How can you tell the difference between a good
cryptography joke and a random string of words?
You can't.  They're indistinguishable.
```

Secure Control Architecture

- Consider the discrete-time system:

$$x[k+1] = f(x[k], u[k]),\ x[0] = x_0,$$
$$y[k] = g(x[k]),$$

- Along with the (nice) dynamic controller:

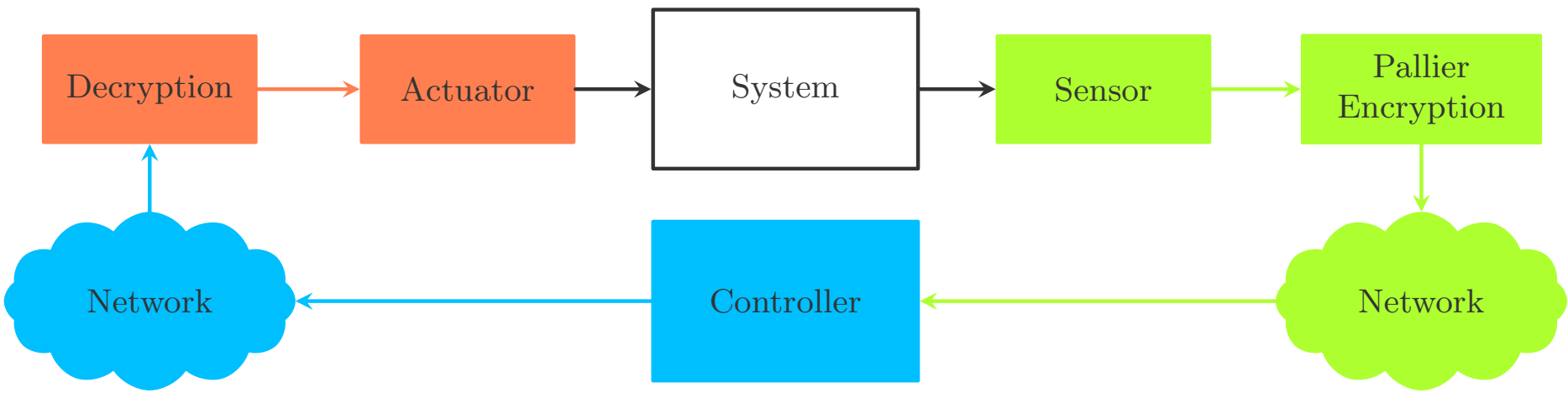
$$x_c[k+1] = Ax_c[k] + B(\overbrace{s[k]}^{reference} - y[k]),\ x_c[0] = x_c[T] = x_c[2T] = \cdots = 0,$$
$$u[k] = Cx_c[k].$$

- The 'periodic reset' makes sure that we don't run out of memory.
- To implement the controller on digital computers one needs to quantise the control parameters and signals.

Assumption: *The controller works well in the presence of quantisation.*

Secure Control Architecture

- The output of the system and control parameters are quantised.
- Let $\bar{*}$ denoted the quantised version of the $*$:
  $\bar{*} = \arg\min_{z \in \mathbb{Q}(n,m)} \|z - *\|_2$.
- Let $\hat{*} = 2^m \bar{*}$ be the lifted version of $\bar{*}$ – integers.
- Let $\tilde{*}$ be the encrypted version of $\hat{*}$ – massive integers.



Secure Control Architecture

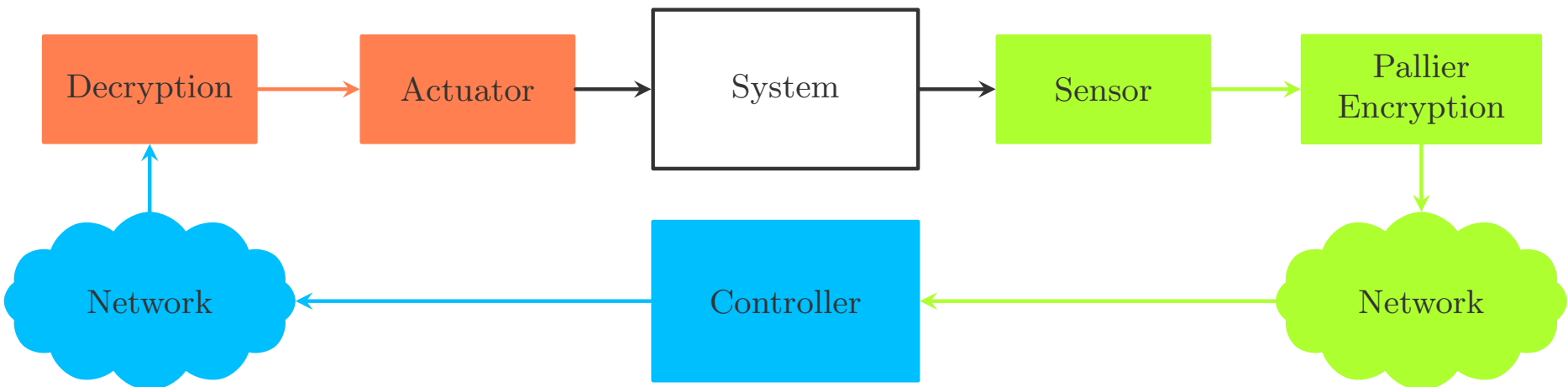
- The controller dynamics in ciphertext ($i = 1, \ldots, n_x$):

$$(\tilde{x}_c)_i[k+1] = \begin{cases} \left[\oplus_{j=1}^{n_x}(\hat{A}_{ij} \otimes (\tilde{x}_c)_j[k])\right] \oplus \left[\oplus_{j=1}^{n_y}(\hat{B}_{ij}[k] \otimes (\tilde{s}_j[k] - \tilde{y}_j[k]))\right], & k+1 \mod T > 0, \\ \mathsf{E}(0, \kappa_p), & k+1 \mod T = 0, \end{cases}$$

$$\tilde{u}_i[k] = \left[\oplus_{j=1}^{n_x}(\hat{C}_{ij} \otimes (\tilde{x}_c)_j[k])\right] \oplus \left[\oplus_{j=1}^{n_y}(\hat{D}_{ij}[k] \otimes (\tilde{s}_j[k] - \tilde{y}_j[k]))\right].$$

$$\hat{u}_i[k] = \mathsf{D}(\tilde{u}_i[k], \kappa_S) \mod 2^{n'},$$

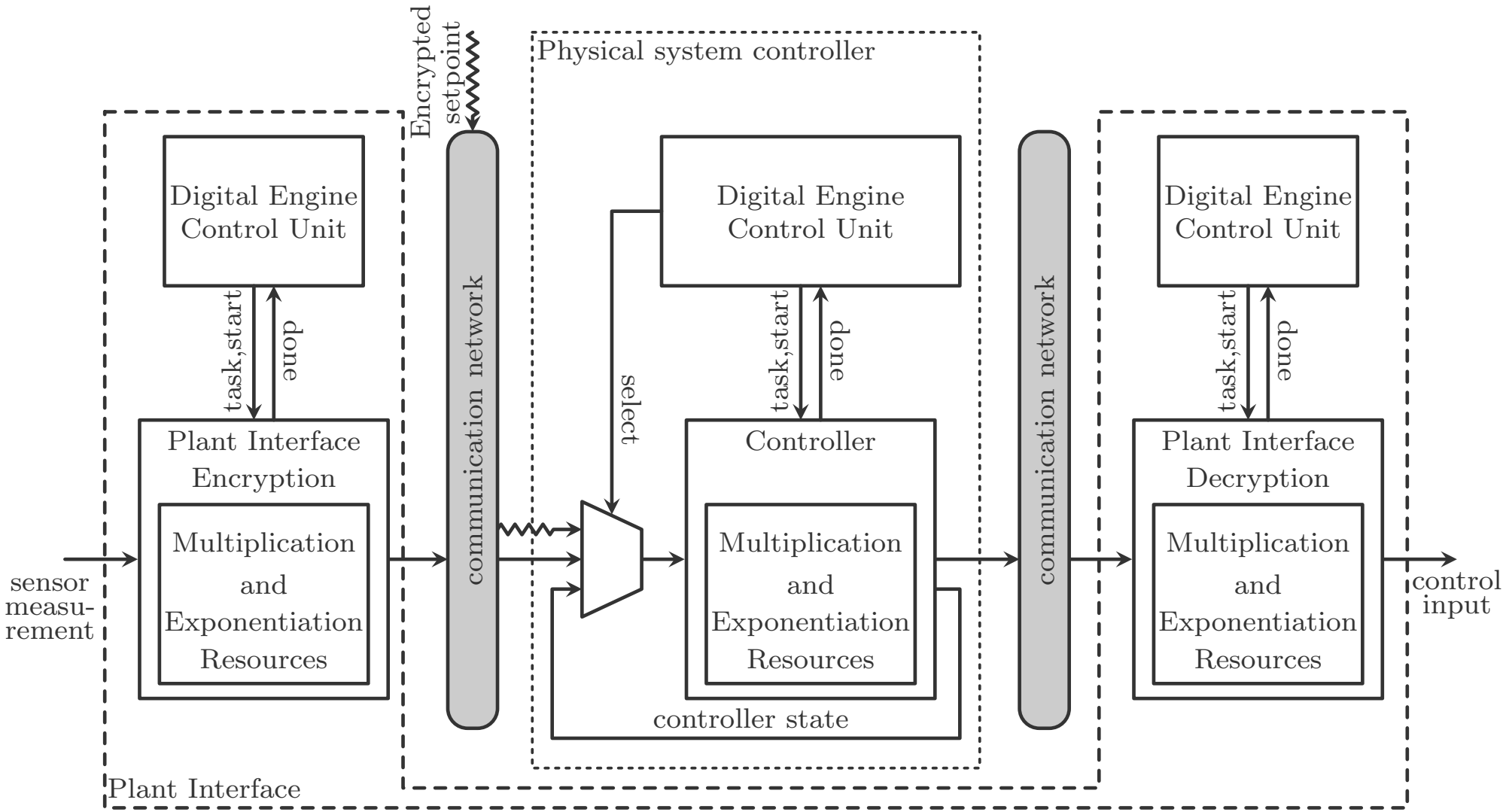
$$\bar{u}_i[k] = 2^{-(k \mod T+2)m}(\hat{u}_i[k] - 2^{n'} 1_{\hat{u}_i[k] \geq 2^{n'}-1}).$$



A lot of arithmatics need to be done. Some certainty about the timing is desired – a custom digital enginer to the rescue. Also, the answer to "how fast is fast enough?" is system dependent.

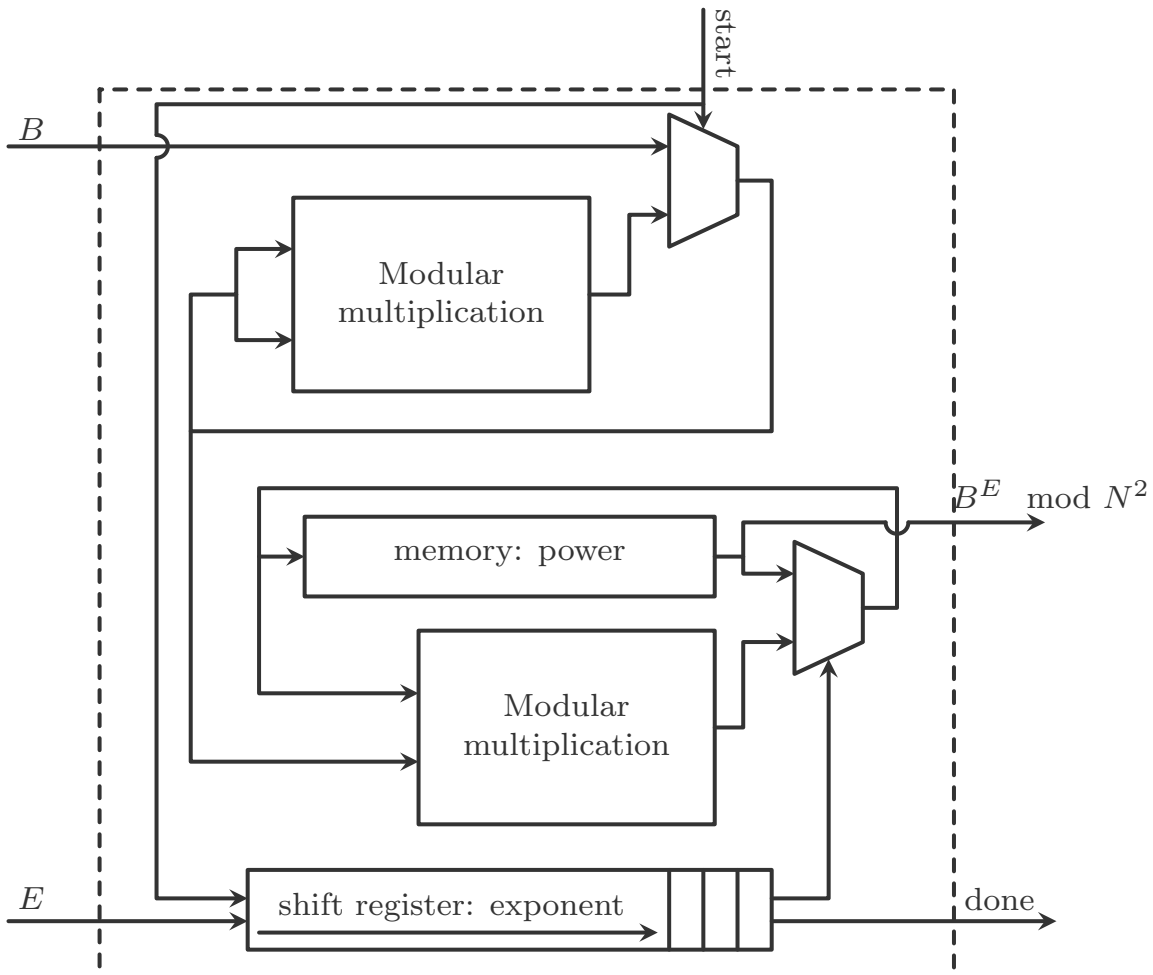
Outline

Secure Control Digital Implementation



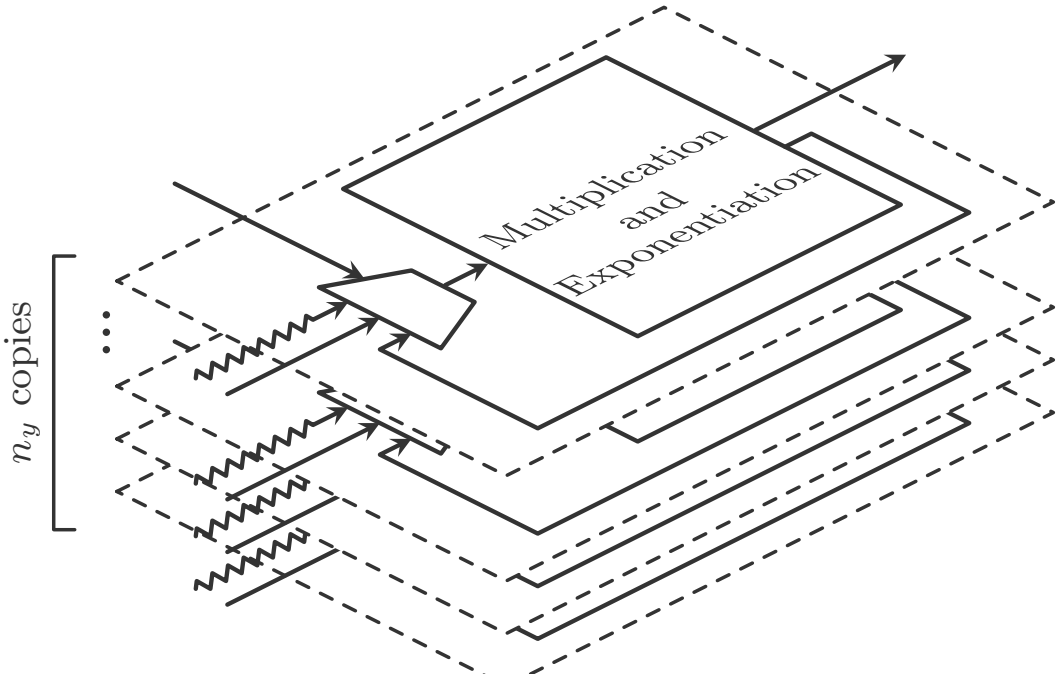
Secure Control Digital Implementation

- The right-to-left binary method for exponentiation involves calculating many sequential modular multiplications.
- Montgomery multiplication is best suited.
- It only involves additions, multiplications, and right shifts.



Secure Control Digital Implementation

- Possible Montie implementation:
  - Karatsuba multiplication-based implementaiton: fast, resource exhaustive.
  - Coarsely Integrated Operand Scanning (CIOS) with a word size of a single bit: can be implemtad by additionions and right shifts.
- Not utilising multi-bit word embedded multipliers available on most modern FPGA devices.
- We use a blockwise implementation of the CIOS method of Montgomery multiplication.
- Elements of the control input can be calculated in parallel:



Outline

Experiment: Stabilising an inverted pendulum

Controller:

$$x[k+1] = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 1} \\ \frac{125\pi}{3072}\begin{bmatrix}500 & 0 & 625\end{bmatrix} & 0 \end{bmatrix} x[k] + \begin{bmatrix} I_{3\times 3} \\ 0_{1\times 3}\end{bmatrix}(s[k]-y[k])$$

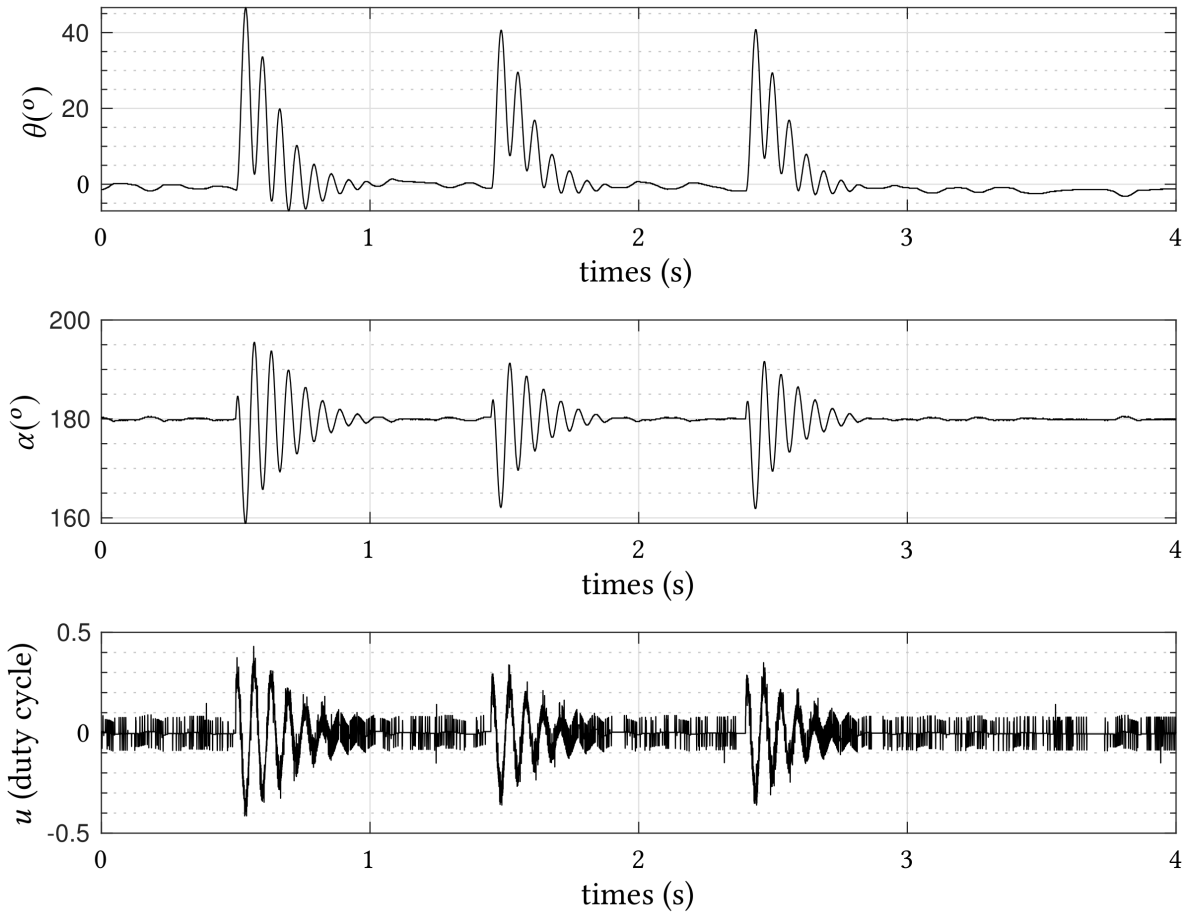
$$u[k] = \begin{bmatrix} \frac{125\pi}{3072}\begin{bmatrix}-500 & -2 & -655\end{bmatrix} & 1\end{bmatrix} x[k],$$

$$s[k] = \begin{bmatrix} 0 \\ \theta_s[k] \\ 1024 \end{bmatrix}, \quad y[k] = \begin{bmatrix} \theta[k] \\ \theta[k] \\ \alpha[k] \end{bmatrix},$$

- sampling frequency of 500 Hz
- control input range of -999 to 999 (duty cycle and direction)
- θ: rotational arm angle
- α: encoder angle with 2^{11} encoder levels
- Encryption key length of 256 bits.
- 32 bits quantisation (7 fractional bits)

Experiment: Stabilising an inverted pendulum

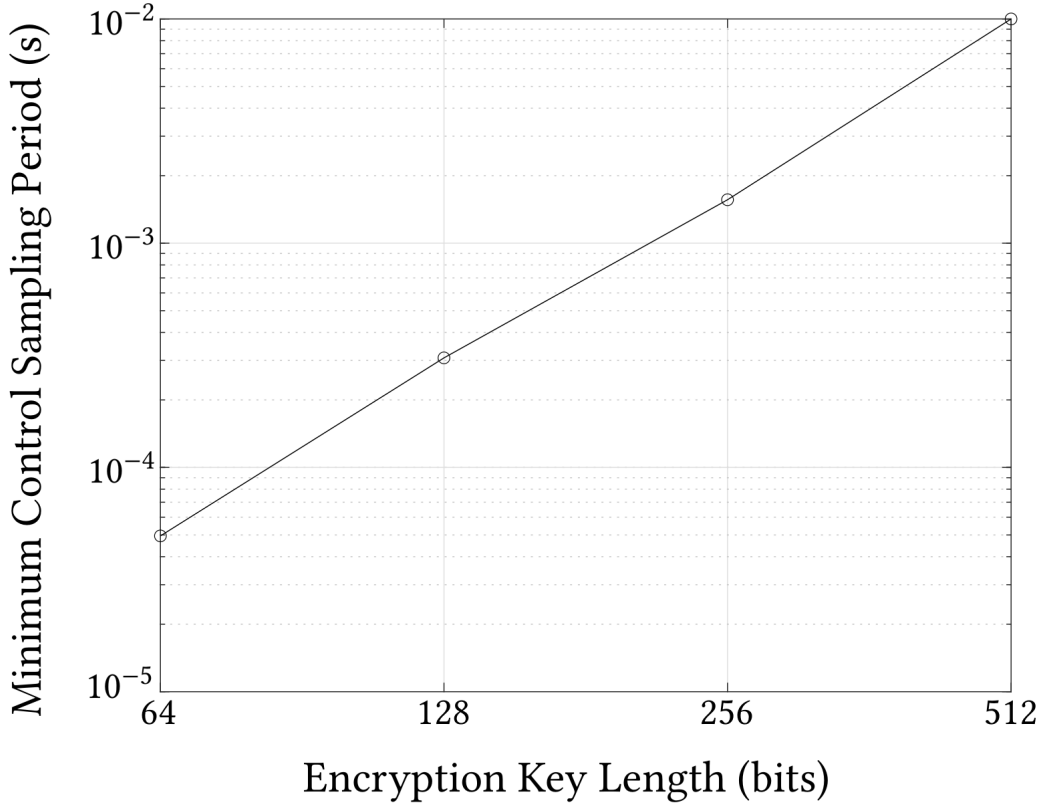
The inverted pendulum system with disturbances introduced at the tip of the pendulum.



Experiment video: `https://youtu.be/ATM0tcecst0`

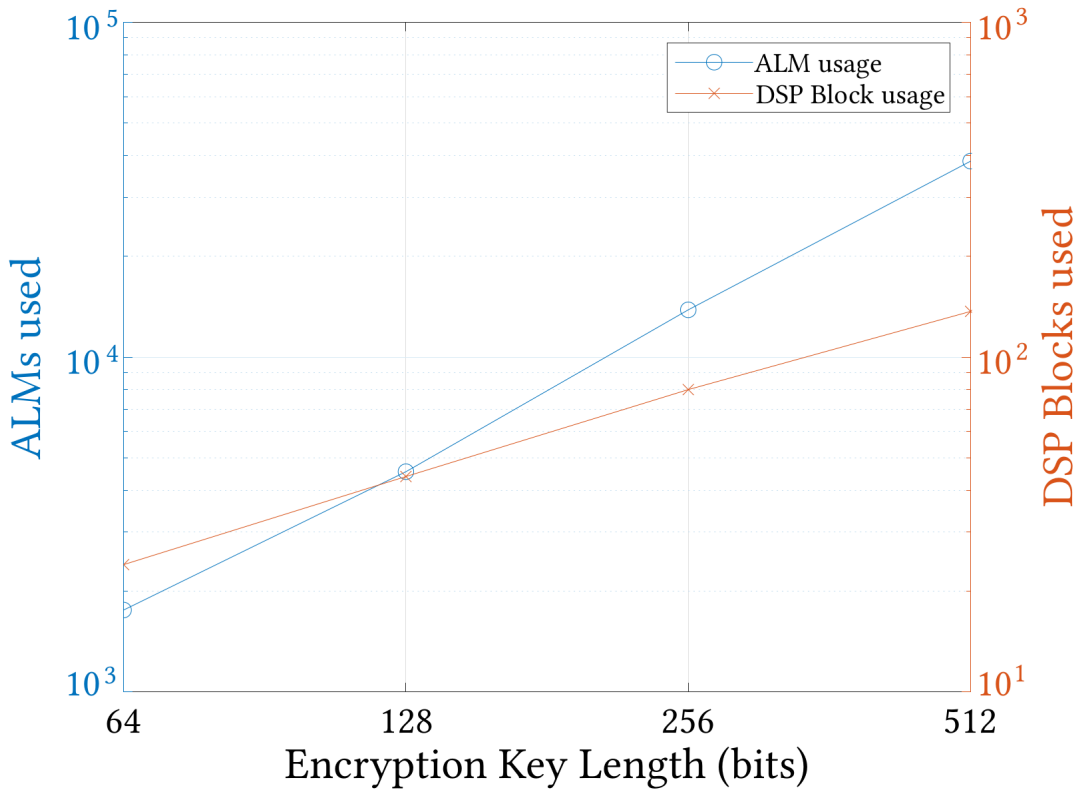
Experiment: Stabilising an inverted pendulum

Minimum control sampling period increases with greater security:



Experiment: Stabilising an inverted pendulum

Usage of hardware resources in the plant interface increases with greater security:



Concluding Remarks and Future Directions

"Oh well I suppose it has come to this."

– Ned Kelly, November 11, 1880, before being hanged at Melbourne Gaol

- A digital implemetation of a semi-homomorphically encrypted control architecture along some experiments were presented.
- HDL code at `https://github.com/availn/EncryptedControl`.
- Design and analysis of encrypted dynamic controllers come with their own challenges, we have recently introduced a framework based on a result by John Moore in the 60's called "fixed-lag smoothing".
- The relationship between the performance and other implementations of Montgomery multiplier of interest.
- The impact of unreliable communication network to be studied.
- Implementing nonlinear control laws is a challenge.
- Making the hardware secure (against Hall effect sensors and power meters)

Thank you! Questions? imanshames.blog