Secrecy Codes for Wireless Control Systems

Anastasios Tsiamis, Konstantinos Gatsis, George J. Pappas

University of Pennsylvania

July 13, 2018
Secrecy Issues in the Internet of Things

- Interconnected sensors and actuators.
- Communication over wireless networks.

The wireless medium may be compromised.

- Broadcast nature.
- Eavesdroppers may intercept sensitive data about the physical system.
Eavesdropping attacks in Dynamical Systems

Dynamical System → Sensor → Channel → User

Eavesdropper

**Goal**
- Protect dynamical state information, e.g. position, velocity.
Eavesdropping attacks in Dynamical Systems

**Goal**
- Protect dynamical state information, e.g. position, velocity.
- Design secrecy codes.
Eavesdropping attacks in Dynamical Systems

Goal
- Protect dynamical state information, e.g. position, velocity.
- Design secrecy codes.

Challenges
- Tradeoff between security and code complexity.
Goal
▶ Protect dynamical state information, e.g. position, velocity.
▶ Design secrecy codes.

Challenges
▶ Tradeoff between security and code complexity.

Approach
▶ Can we exploit model knowledge/physics for secrecy?
Previous work

Encryption

- Adversary should have bounded computational capabilities.
- Communication/computation overheads.

Our approach: State-Secrecy Codes

- Exploit the system dynamics.
- New tradeoff between code complexity and security.
- Simple and fast.
- Strong security guarantees about the current state.

Related work

- Codes for several types of linear systems
  A. Tsiamis et al. CDC 2017, ACC 2018, CDC 2018
- Non-coding approach
  A. S. Leong et al. IFAC 2017, CDC 2017
  A. Tsiamis et al. IFAC 2017
Previous work

Encryption
- Adversary should have bounded computational capabilities.
- Communication/computation overheads.

Physical layer security
- Provable guarantees.
- Requires eavesdropper’s channel model.
Previous work

Encryption

- Adversary should have bounded computational capabilities.
- Communication/computation overheads.

Physical layer security

- Provable guarantees.
- Requires eavesdropper’s channel model.

Our approach: State-Secrecy Codes

- Exploit the system dynamics.
- New tradeoff between code complexity and security.
- Simple and fast.
- Strong security guarantees about the current state.
Previous work

Our approach: State-Secrecy Codes

- Exploit the system dynamics.
- New tradeoff between code complexity and security.
- Simple and fast.
- Strong security guarantees about the current state.

Related work

- Codes for several types of linear systems
  A. Tsiamis et al. CDC 2017, ACC 2018, CDC 2018
- Non-coding approach
  A. S. Leong et al. IFAC 2017, CDC 2017
  A. Tsiamis et al. IFAC 2017
Model

System $x_k$ → Sensor & Encoder → ACK → User $\hat{x}_{u,k}$

- $z_k$ encoded version of state $x_k$
- $\gamma_{u,k}$
- $\gamma_{e,k}$

Eavesdropper $\hat{x}_{e,k}$

Linear System

- state $x_k$; e.g. position and velocity at time $k$

$$x_{k+1} = Ax_k + w_{k+1}, \quad w_k : \text{Gaussian process noise}$$
Model

Linear System

- state $x_k$; e.g. position and velocity at time $k$
  
  $$x_{k+1} = Ax_k + w_{k+1}, \quad w_k : \text{Gaussian process noise}$$

Sensor and encoder

- $z_k$ encoded version of state $x_k$. 
Model

Channel

- Packet drop channels:
  - Message either received intact or dropped.
  - $\gamma_{u,k} = 1$: message $z_k$ received.
  - $\gamma_{u,k} = 0$: message $z_k$ dropped.
  - Acknowledgment signals available.
Decoders: Minimum Mean Square Error Estimators

- Information sets:

\[ \mathcal{I}_{u,k} = \{ \text{Received messages } z_k \} \]
\[ \mathcal{I}_{e,k} = \{ \text{Intercepted messages } z_k, \text{User’s outcomes } \gamma_{u,k} \} \]
Decoders: Minimum Mean Square Error Estimators

- Information sets:
  \[ \mathcal{I}_{u,k} = \{ \text{Received messages } z_k \} \]
  \[ \mathcal{I}_{e,k} = \{ \text{Intercepted messages } z_k, \text{User's outcomes } \gamma_{u,k} \} \]

- Estimation:
  \[ \hat{x}_{u,k} \text{ minimizes } \mathbb{E}(\| x_k - \hat{x}_{u,k} \|^2 | \mathcal{I}_{u,k}) \]
  \[ P_{u,k} \text{ user mmse covariance given } \mathcal{I}_{u,k} \]
**Problem: Secrecy**

<table>
<thead>
<tr>
<th>Secrecy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find a coding scheme such that:</td>
</tr>
<tr>
<td>▶ <strong>Eavesdropper</strong>’s minimum mean square error (mmse) for the current state is <strong>maximum</strong> asymptotically.</td>
</tr>
<tr>
<td>▶ <strong>User</strong>’s mmse is <strong>optimal</strong>.</td>
</tr>
</tbody>
</table>

**Assumptions**

- Public $A$.
- Passive eavesdropper.
- Eavesdropper knows model, coding scheme, acknowledgments.
State-Secrecy Code

Coding Scheme

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

\( t_k \): the most recent message received at the user

- Matrix \( L \) depends on the dynamical system’s model.

- Choice of \( L \): makes \( z_k \) less correlated to \( x_k \) for the eavesdropper.

- ACKs: user and sensor agree on \( t_k \).

- Fast and simple.

- Can be used along with encryption.
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

Encoder | User | Eavesdropper
--- | --- | ---
\( x_0 \) |  |  

User decodes by adding.
Eavesdropper decodes initially.
Eavesdropper cannot decode for \( k > 2 \).
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✔️</td>
<td>( x_0 )</td>
</tr>
</tbody>
</table>
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

Encoder

User

Eavesdropper

\[ x_0 \]

✓

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]

\[ x_0 \]
### Example

**Code**

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✔</td>
<td>( x_0 )</td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>✗</td>
<td>—</td>
</tr>
</tbody>
</table>
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✔️</td>
<td><strong>x_0</strong></td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>✗</td>
<td>-</td>
</tr>
<tr>
<td>( x_2 - L^2x_0 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

User decodes by adding. Eavesdropper decodes initially. Eavesdropper cannot decode for \( k > 2 \).
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td>( x_2 - L^2x_0 )</td>
<td>✔️</td>
<td>( x_2 - L^2x_0 )</td>
</tr>
</tbody>
</table>
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✓</td>
<td>( x_0 )</td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>✗</td>
<td>-</td>
</tr>
<tr>
<td>( x_2 - L^2x_0 )</td>
<td>✓</td>
<td>( x_2 - L^2x_0 )</td>
</tr>
<tr>
<td>( x_3 - Lx_2 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>( x_2 - L^2x_0 )</td>
<td>✓</td>
<td>( x_2 - L^2x_0 )</td>
</tr>
<tr>
<td>( x_3 - Lx_2 )</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✓</td>
<td>( x_0 )</td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>✗</td>
<td>-</td>
</tr>
<tr>
<td>( x_2 - L^2x_0 )</td>
<td>✓</td>
<td>( x_2 - L^2x_0 )</td>
</tr>
<tr>
<td>( x_3 - Lx_2 )</td>
<td>✗</td>
<td>-</td>
</tr>
<tr>
<td>( x_4 - L^2x_2 )</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

User decodes by adding.
Eavesdropper decodes initially.
Eavesdropper cannot decode for \( k > 2 \).
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td></td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>( x_2 - L^2x_0 )</td>
<td>( \checkmark )</td>
<td>( x_2 - L^2x_0 )</td>
</tr>
<tr>
<td>( x_3 - Lx_2 )</td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>( x_4 - L^2x_2 )</td>
<td>( \checkmark )</td>
<td>( x_4 - L^2x_2 )</td>
</tr>
</tbody>
</table>
## Example

### Code

\[
z_k = x_k - L^{k-t_k} x_{t_k}
\]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_0)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(x_1 - Lx_0)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>(x_2 - L^2x_0)</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>(x_3 - Lx_2)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>(x_4 - L^2x_2)</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

User decodes by adding.

Eavesdropper decodes initially.

Eavesdropper cannot decode for \(k > 2\).
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>( x_1 - L x_0 )</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>( x_2 - L^2 x_0 )</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>( x_3 - L x_2 )</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>( x_4 - L^2 x_2 )</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- User decodes by adding.

_Example_

_Encoder_  _User_  _Eavesdropper_

\( x_0 \)  ✓  x_0  ✓  x_0

\( x_1 - L x_0 \)  ✗  —  ✓  x_1 - L x_0

\( x_2 - L^2 x_0 \)  ✓  x_2 - L^2 x_0  ✗  —

\( x_3 - L x_2 \)  ✗  —  ✓  x_3 - L x_2

\( x_4 - L^2 x_2 \)  ✓  x_4 - L^2 x_2  ✓  x_4 - L^2 x_2

▶ User decodes by adding.
## Example

### Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_0)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(x_1 - Lx_0)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>(x_2 - L^2 x_0)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(x_3 - Lx_2)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>(x_4 - L^2 x_2)</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- User decodes by adding.
- Eavesdropper decodes initially.
- Eavesdropper cannot decode for \(k > 2\).
Example

Code

$$z_k = x_k - L^{k-t_k} x_{t_k}$$

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0$</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>$x_1 - Lx_0$</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>$x_2 - L^2x_0$</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>$x_3 - Lx_2$</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>$x_4 - L^2x_2$</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

- User decodes by adding.
- Eavesdropper decodes initially.
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>( x_0 - Lx_0 )</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>( x_2 - L^2x_0 )</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>( x_2 - Lx_2 )</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>( x_4 - L^2x_2 )</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- User decodes by adding.
- Eavesdropper decodes initially.
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>( x_2 - L^2 x_0 )</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>( x_3 - Lx_2 )</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>( x_4 - L^2 x_2 )</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- User decodes by adding.
- Eavesdropper decodes initially.
- Eavesdropper cannot decode for \( k > 2 \).
Example

Code

\[ z_k = x_k - L^{k-t_k} x_{t_k} \]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>User</th>
<th>Eavesdropper</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>( x_1 - Lx_0 )</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>( x_2 - L^2x_0 )</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>( x_3 - Lx_2 )</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>( x_4 - L^2x_2 )</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Critical event

When the user receives while the eavesdropper misses:

\[ \gamma_{u,k} = 1, \gamma_{e,k} = 0, \text{ for some } k \]

Damages the eavesdropper (here for \( k = 2 \)).
Theorem 1: Secrecy

If the critical event occurs at time $k_0$ secrecy is achieved:

- **Eavesdropper’s** mmse is asymptotically maximum:
  
  $$P_{e,k} \rightarrow \text{maximum value}$$

- **User’s** mmse remains optimal:
  
  $$P_{u,k} = 0, \text{ when user receives. } (\gamma_{u,k} = 1)$$

- One occurrence of the critical event is sufficient.
  (user receives and eavesdropper fails to intercept at some time $k$.)

- Condition holds for most packet dropping channels.
Simulation scenario

- Linearized dynamics, planar quadrotor
- Hovers around target point \((0, 0, 0)\)
- State is position and velocity
Simulation scenario

- Height estimation
Simulation scenario

- Height estimation
- Critical event occurs at time $k = 47$
Simulation scenario

- Height estimation

- Critical event occurs at time $k = 47$

- Eavesdropper gets confused about the state; trivial estimate $x_e = 0$
Simulation scenario

- Height estimation
- Critical event occurs at time $k = 47$
- Eavesdropper gets confused about the state; trivial estimate $x_e = 0$
- User’s estimation remains accurate
Conclusion

Summary

► We can exploit the dynamical system model/physics for secrecy.
► Codes specialized for dynamical systems.
► Simple and fast, strong guarantees for the current state.
► Secrecy guarantees for several types of linear systems.
Conclusion

Summary

▶ We can exploit the dynamical system model/physics for secrecy.
▶ Codes specialized for dynamical systems.
▶ Simple and fast, strong guarantees for the current state.
▶ Secrecy guarantees for several types of linear systems.

Future Work

▶ How to defend against active eavesdroppers?
▶ How can we combine codes with encryption efficiently?
▶ How can we apply the codes in closed-loop control systems?
▶ How can we protect past states?
Thank you!