Establishing Data Integrity in Networks of Cyber-Physical Systems

Mihalis Tsiakkas*, Panayiotis Kolios, Marios Polycarpou and Christos Panayiotou

Abstract— **Cyber-physical systems are pervasive in today's society with applications ranging from critical infrastructure systems such as the power grid or water distribution network to industrial control systems. It is clear that the reliability, safety and security of such systems is imperative. In this paper we propose utilizing tools borrowed from cryptography to ensure that false data injection attacks affecting the communications of cyber-physical systems are effectively and reliably detected. A practical example is presented to both motivate and validate the developed tools.**

I. Introduction

Recent years have seen a multitude of attacks on cyber-physical systems; some with devastating effects. Prevalent examples include the StuxNet worm, that targeted and successfully destroyed several uranium enrichment centrifuges in Iran [1], as well as the Ukrainian power outage incident in which thousands of people were left without an electrical supply for an extended period of time [2]. Such attacks can have a significant socio-economic impact and even lead to loss of human life if for example a hospital is targeted.

Many attack detection schemes have been proposed by the control community following the rise of such incidents, many utilizing Luenberger type observers [3], [4], [5], [6]. For example, Pasqualetti et. al [3] suggest one approach to using such observers in order for detecting and possibly identifying attacks. However, the assumptions imposed on the attack significantly impact the practical applicability of the proposed methods. Furthermore, it is simple to show that in the framework being considered, one could find infinitely many undetectable attacks as defined therein. In the context of [3] a data injection attack is said to be undetectable if it does not affect the output of the plant, or equivalently if it exploits the zero dynamics of the plant.

Another drawback of such observer based methods is the inherent need for a threshold, where if the residual is below some predefined value the perturbation is not classified as an attack. This allows an attacker to persistently alter the state measurements while remaining undetected. Even if this does not destabilize the plant or network, it could have a detrimental impact on its operating performance and efficiency. This would constitute an $(\epsilon, \mathbf{s})$-effective attack as defined by [4]; it was proven therein that such an attack always exists. An attack is called $(\epsilon, \mathbf{s})$-effective if for some sensor subset $\mathbf{s}$ it leads to increased estimation error in comparison to the optimal observer and in the absence of an attack.

A different type of attack is proposed in [7], termed "perfect", which attempts to destabilize the observer residual without altering the output of the observer itself. This type of attack is addressed by [8] using a time varying coding scheme in the form of a pre-shared, secret and bijective mapping to "encrypt" the transmitted sensor measurement data. It should be noted that as stated by [9], such attacks are practically infeasible due to the high amount of information required by the adversary as well as the fact that any modeling errors would quickly expose any such attempts.

A more realistic approach to modeling a cyber-attack is that taken in [10] were no significant assumptions are imposed on the trajectory of the attack signal. Instead, the attack is only limited by its cardinality (the number of sensors that are corrupted). An error correction approach is then followed using the tools developed by [11] in order to recover the true (unaltered) measurement vector. The method proposed therein however, requires the solution of a linear program at each iteration the necessary computational power for which might be excessive for certain applications.

In this paper a validation method for networks of cyber-physical dynamic systems is presented aiming to establish the authenticity and integrity of transmitted data. This is achieved using a dynamically updated, secret key in combination with cryptographic hash functions. The key is updated using previous state information from the source plant that has already been transmitted and validated as true. A cryptographic hash function is used to generate a hash value for the pair of state measurement vector and generated key, which is the transmitted over the network along with the measurement. This calculation is performed again at the destination node and the two hash values are compared with a mismatch indicating that the data has been tampered with.

The proposed method is a simple, yet effective approach to deterring false data injection attacks. Some advantages offered over other attack detection techniques proposed in the literature are:

1) The detection of attacks is absolute, meaning that even infinitesimal perturbations from an attacker would be detected. This is unlike previously discussed methods which necessarily allow for a threshold; thus enabling an adversary to "hide" an attack by reducing its magnitude to within noise levels. In fact, measurement noise helps improve the security of the proposed scheme by increasing the generated key entropy. Furthermore, an alarm is raised only if an attack has occurred.
2) A fault would not be mistaken for an attack. The

The authors are with the KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, Cyprus. m.tsiakkas@ieee.org, pkolios@ucy.ac.cy, mpolycar@ucy.ac.cy, christosp@ucy.ac.cy

* Corresponding author.

proposed method is implemented after sensor measurements are taken; therefore the authenticity and integrity of a message can be positively established even if the hardware itself is faulty. This allows for differentiating between faults and attacks.

3) The proposed solution is fully agnostic of the dynamics of each node (even though it is affected by them) as well as the structure of the underlying communication graph. As a result, no significant assumptions are imposed. The only assumption is that the underlying communication graph is weakly connected which does not affect the applicability of the presented technique in any situation.

The proposed method is only able to detect a false data injection attack; not prevent one, leading to a data availability problem. The theory developed in [12] could then be used to stabilize the system if the appropriate conditions are met. Alternatively, if the communication graph contains other directed paths from the source to the destination node whose integrity has been established, the measurement data could be re-routed with the problem being reduced to that of a simple time delay. Such a case is presented herein.

This paper is organized as follows. A brief introduction to hash functions is given in section II. This is followed by a more formal formulation of the problem being considered and a motivating example in section III. A solution is then proposed in section IV, including examples of how to select the various parts of the attack detection scheme. Subsequently, the motivating example is revisited in section V and a successful application of the proposed method is demonstrated. Finally, a few concluding remarks are given in section VI.

II. Brief introduction to hash functions

Hash functions are a widely used tool with applications ranging from data validation to secure password storage. The purpose of a hash function is to generate a fixed length identifying value for a given data set. For example, given a set of integers $\{m_1, \ldots, m_n\} \subseteq \mathbb{N}$, one could define a hash as $\sum_{i=1}^{n} |m_i| \mod p$ for some $p \in \mathbb{N}_+$. Clearly, this approach has some significant drawbacks one of which is the fact that replacing any element m_i in the original set with $m_i + p$ would not affect the resulting hash value. Eliminating such collisions is impossible; however, cryptographic hash functions have been developed so that finding a collision is very computationally expensive to the point of practical infeasibility. An ideal cryptographic hash function is one that is injective, fast to compute, with significant output variation for small input deviations and impossible to invert.

Cryptographic hash functions are commonly used in two modes. The first is as Modification Detection Codes (MDCs) where the output is used to verify that the correct data was transmitted. The second mode is as Message Authentication Codes (MACs) where an additional secret key is used so that not only is integrity verified but also its source. Keyed-hash message authentication code (HMAC) was first outlined in [13] utilizing the MD5 or SHA-1 hashing algorithms.

A useful property of several hashing algorithms is that they are $\mathcal{O}(n)$, that is the time required to compute the hash of a data set is a linear function of the size of that set. This is a result of the fact that many hash functions are based on block ciphers [14] that operate on fixed-length data sets (blocks) by performing a constant number of operations.

An in depth treatment of hash function and their applications can be found in [14].

III. Problem Formulation

Consider a directed graph (digraph) $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where each vertex $P_i \in \mathcal{V} : \mathbb{R}^q \mapsto \mathbb{R}^p$ denotes an arbitrary, possibly nonlinear dynamical system. Every edge $e_{i,j} = \{P_i, P_j\} \in \mathcal{E}$ implies that the state vector (or part thereof) of the system at $P_i \in \mathcal{V}$, denoted $x_i \in \mathbb{R}^{n_i}$, is communicated over the network to $P_j \in \mathcal{V}$ where it can be used for any necessary purposes including control and monitoring; hence state measurements from P_i are transmitted to all its out-neighbors in $\mathcal{N}_-(P_i) = \{P_j \in \mathcal{V} : \{P_i, P_j\} \in \mathcal{E}\}$[1].

We suppose that $\mathcal{G}$ is at least weakly connected; this means that for any two vertices in $\mathcal{V}$, there exists a sequence of incident edges[2] (ignoring directionality) connecting the two. Failing this supposition implies that the graph can be decomposed into two disjoint subgraphs that share no information and could therefore be studied independently. For more information on the subject of Graph theory and related concepts, the reader is referred to [15].

As is the case with any communication channel, there are bound to be vulnerabilities in the transmission of the state measurements. These could potentially allow a malicious attacker to substitute measurement data in order to achieve some goal; be it destabilization of the network or hide some other malicious activity. Our aim is to prevent such an occurrence by validating the data received at each vertex.

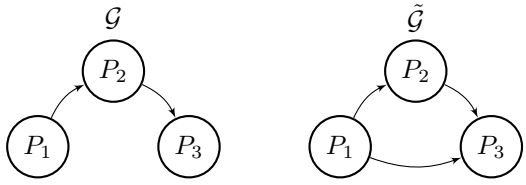


Fig. 1: Connection example.

For the purposes of this paper, the graph being considered does not correspond to the physical interconnection of nodes in the network, but rather the cyber connections where data can be *directly* transmitted (i.e. not routed via other vertices). To see the necessity of direct connections only, consider the two graphs shown in Figure 1. $\mathcal{G}$ represents the graph of direct connections between three nodes; while $\tilde{\mathcal{G}}$ shows the possible data sharing connections between the same nodes, where P_3 obtains data from P_1 routed via P_2. Now suppose that the edge $e_{1,2}$ or $e_{2,3}$ in $\mathcal{G}$ is compromised in some sense;

[1]The in-neighborhood of P_i is similarly defined as $\mathcal{N}_+(P_i) = \{P_j \in \mathcal{V} : \{P_j, P_i\} \in \mathcal{E}\}$.

[2]Edges in a graph are termed incident if they connect to the same vertex.

it then follows that the edge $e_{1,3}$ in $\tilde{\mathcal{G}}$ is also implicitly compromised and thus any received data cannot be trusted.

Once measures are put in place to establish the integrity and authenticity of the messages being communicated, then indirectly transmitted data can be used as necessary. One such example will be given at the end of the present paper.

A. Motivating Example

A motivating example will now be presented to demonstrate how a man-in-the-middle false data injection attack can have detrimental effects on the control of robot formations. Consider three mobile ground vehicles with a communication topology induced by the digraph shown in Figure 2.

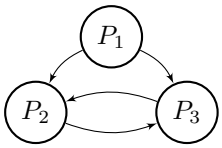


Fig. 2: Example communication network among three autonomous agents.

Furthermore, suppose that each robot is locally controlled using a linearization strategy (for example feedback linearization or sliding mode control) resulting in the closed loop velocity dynamics[3]

$$v_i^{k+1} = \begin{bmatrix} 0.741 & 0.008 \\ -0.066 & 0.809 \end{bmatrix} v_i^k + \begin{bmatrix} 0.407 & 0.051 \\ 0.453 & 0.365 \end{bmatrix} u_i^k, \quad (1)$$

where $u_i^k \in \mathbb{R}^2$ and $v_i^k \in \mathbb{R}^2$ correspond to the reference and actual velocities of P_i at the time index $k \in \bar{\mathbb{N}}_+$ respectively. The position $x_i^k \in \mathbb{R}^2$ of the same robot is given by

$$x_i^{k+1} = x_i^k + v_i^k h, \quad (2)$$

where $h \in \mathbb{R}_+$ is the sampling time of the system.

We now assume the following hardware configuration. None of the vehicles is equipped with an absolute positioning system. However, P_1 (which is the formation leader) is able to measure the relative position (distance and direction) of each of the two other vehicles with respect to itself. This information is transmitted to the relevant vehicle over the edges $e_{1,2}$ and $e_{1,3}$.

With the x and y axes being horizontal and vertical to the plane of the page respectively, the desired formation is for P_1 to lead with P_2 and P_3 lagging behind with 1 unit difference in the x direction and ± 1 unit in the y direction. A sinusoidal reference velocity signal is given to the leader while P_2 and P_3 are tasked with maintaining their position relative to P_1. Simulation results of both the nominal case and when the system is under attack are shown in Figure 3. Figures 4a and 4b show the time evolution of the x and y positions of the robots respectively.

In the attack scenario, the adversary additively perturbs the relative position measurement in the y direction transmitted from P_1 to P_2 over the edge $e_{1,2}$ in an attempt to cause a collision between P_2 and P_3. The attack can be summarized by the difference equations

$$\kappa^{k+1} = \kappa^k + \hat{\kappa} \text{ and} \quad (3a)$$

$$\hat{\Delta}_{1,2}^k = \Delta_{1,2}^k + \begin{bmatrix} 0 \\ \kappa^k \end{bmatrix}, \quad (3b)$$

where $\Delta_{1,2}^k \in \mathbb{R}^2$ is the relative position measurement between P_1 and P_2 as transmitted by P_1 over the edge $e_{1,2}$ and $\hat{\Delta}_{1,2} \in \mathbb{R}^2$ is its corrupt equivalent received by P_2. The attack signal is given by $\kappa^k \in \mathbb{R}$ with $\hat{\kappa} \in \mathbb{R}$ being a constant that determines how quickly the collision will occur.

As part of the nominal mission of the robots, P_2 and P_3 are always at the same x position with a two unit difference in the y direction. It is a provable fact that by additively perturbing $\Delta_{1,2}^k$ only in the y direction, a collision is guaranteed to occur. It can be seen from the provided simulation results that the adversary's goal is achieved with a collision occurring at approximately $k = 50$. This example is revisited at the end of this paper.

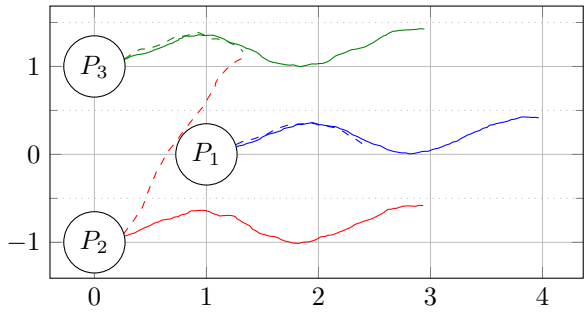
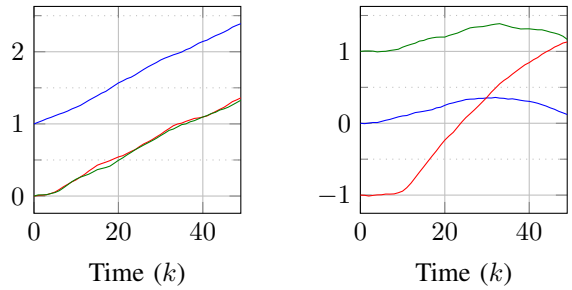


Fig. 3: $x - y$ position plot of robot formation control simulation results under normal operation (solid lines) and under false data injection attack (dashed lines).



(a) x position (b) y position

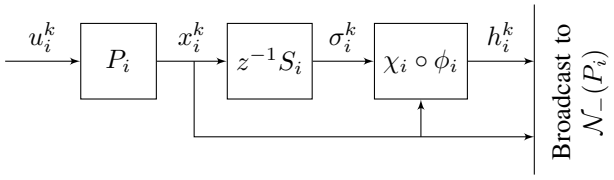
Fig. 4: Time evolution of robot positions under false data injection attack.

IV. Proposed Solution

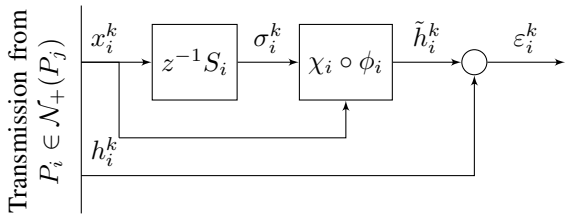
A. Overview

The proposed solution makes use of simulated artificial dynamic systems at each vertex in the graph, knowledge of which is shared across its out-neighborhood but kept secret from third parties. These will henceforth be referred to as hash key dynamics since their output will be used as a key in generating a HMAC. Figure 5 depicts a high level block diagram representation of the suggested strategy.

[3]Homogeneity and low order of vertex dynamics are imposed to simplify the simulation. This does not affect the validity of the presented results.



(a) Hash generator at the vertex $P_i \in \mathcal{V}$



(b) Hash monitor at vertex $P_j \in \mathcal{N}_-(P_i)$

Fig. 5: Block diagram of proposed method. z^{-1} represents the unit delay operator (z-transform) while $\chi_i \circ \phi_i$ denotes the composition of the functions χ_i and ϕ_i.

In the above figures, $S_i : \mathbb{R}^{n_i} \mapsto \mathbb{N}_+$ is the discrete time system corresponding to the hash key dynamics of $P_i \in \mathcal{V}$. S_i takes as input the past state measurement vector x_i^{k-1} and outputs a time varying value $\sigma_i^k \in \mathbb{N}_+$, where $k \in \bar{\mathbb{N}}_+$ denotes the iteration variable. The past measurement vector is used since it is assumed that it was verified in the previous transmission, thus preventing possibly corrupted data from affecting the hash-key dynamics.

The latest measured state vector x_i^k and generated hash key σ_i^k are then passed through a data aggregator function $\phi_i : \mathbb{R}^{n_i} \times \mathbb{N}_+ \mapsto \mathbb{N}_+$. The purpose of this function is to merge this data into a single positive integer value that can be hashed using the algorithm of choice.

Finally, the output of ϕ_i is processed to generate a hash value using the hash function $\chi_i : \bar{\mathbb{N}}_+ \mapsto \mathbb{N}_+$. The pair $\{x_i^k, h_i^k\}$ consisting of the measured state vector and generated hash value, is then broadcast over the graph to all the out-neighbors of P_i in $\mathcal{N}_-(P_i)$.

At the receiving/monitoring vertices $P_j \in \mathcal{N}_-(P_i)$ a parallel procedure is performed. When $\{x_i^k, h_i^k\}$ are received, the output of S_i is updated to obtain σ_i^k. The expected hash $\tilde{h}_i^k$ is then calculated and compared against the received hash value h_i^k. If $h_i^k \neq \tilde{h}_i^k$ an alarm is raised, labeled ε_i^k in Figure 5b, to either alert operators or trigger some predefined defense procedure.

An interesting aspect of the proposed method is the exchange of necessary information between the vertices of the graph; for example the parameters of S_i. This could be done *a priori* with the data hard-coded at each node. Alternatively, before the control system becomes operational, the Diffie-Hellman key exchange protocol [16] could be used to establish a secure/encrypted communication channel over which the parameters can be agreed. Note that this channel could not be used for control purposes as encryption imposes significant computational overheads and would thus drastically increase the sampling time.

In contrast to many cases in control theory, measurement noise is beneficial and enhances the security of the method. This follows by noting that disturbances are included in the sensor measurements which are used at both nodes. Hence the same key is generated regardless of the magnitude of noise; furthermore the added randomness leads to increased entropy in the hash key. The same argument also implies that faults do not cause a hash mismatch allowing the proposed technique to differentiate between faults and attacks (if used in conjunction with more traditional fault detection methods).

The main contribution of this paper is the overall attack detection methodology as presented in Figure 5. In the following subsections, we will discuss in more detail the various parts of this scheme, specifically how ϕ_i and S_i can be selected. It should be noted that there are many possible alternatives and one should construct ϕ_i and S_i in such a way that is suitable for the system being considered.

B. Selecting ϕ_i

The choice of the data aggrerator ϕ_i function can have a significant impact on the security of the proposed method.

The most important property required from ϕ_i is that it is injective, since otherwise the advantages of using a cryptographic hash function are rendered void. To see this consider for example the function $\phi_i(x_i, \sigma_i) = \lceil \|x_i\| \rceil + \sigma_i$. Since $\|x_i\| = \|Ux_i\|$ for any unitary matrix $U \in \mathbb{R}^{n_i \times n_i}$, it follows that $\chi_i \circ \phi_i(x_i, \sigma_i) = \chi_i \circ \phi_i(Ux_i, \sigma_i)$.

A reasonable choice for this data aggregator function would be mapping to $\bar{\mathbb{N}}_+$ using the Unicode standard followed by concatenation using some predefined delimiter character. Alternatively, a more mathematical approach could be taken as follows. First define the function $\zeta : \mathbb{R} \mapsto \bar{\mathbb{N}}_+$ as

$$\zeta(\alpha) = \upsilon(\alpha) \| \upsilon \circ \xi(\alpha) \| \xi(\alpha) \Big\| \Big\lfloor 10^{m - \xi(\alpha) - 1} |\alpha| \Big\rfloor \tag{4}$$

where $m \in \mathbb{N}_+$ denotes the number of digits being encoded, $a \| b$ is the concatenation of $a, b \in \mathbb{N}_+$, $\xi(\alpha) = \lfloor \log_{10} |\alpha| \rfloor$ and $\upsilon : \mathbb{R} \mapsto \{0, 1\}$ corresponds to the step function defined as $\upsilon(\alpha) = \{0 \text{ if } \alpha \leq 0, 1 \text{ otherwise}\}$. An immediate problem that could arise from the above is the singularity at $\xi(0)$; this however can be handled simply by definition. Then a candidate for ϕ_i can be defined as

$$\phi_i \left(x_i^k, \sigma_i^k \right) = \sigma_i^k \| \zeta \left(x_{i,1}^k \right) \| \cdots \| \zeta \left(x_{i,n_i}^k \right), \tag{5}$$

where $x_{i,j}^k \in \mathbb{R}$ corresponds to the j^{th} element of x_i^k.

The advantage of using the above over the aforementioned Unicode approach is that it results in significantly less data to be hashed and therefore improves the efficiency and possible transmission rate.

C. Selecting S_i

A multitude of functions can be used to generate the hashing key, for example one possibility is to use the congruential generators discussed in [17]. In fact, even a simple linear, time-invariant model based approach could be taken. Here however we propose a method based on the Blum-Blum-Shub (BBS) pseudorandom number generator (PRNG) which can be simply expressed as follows [18].

Let $m \in \mathbb{N}_+$ be the product of two sufficiently large prime numbers $p \in \mathbb{N}_+$ and $q \in \mathbb{N}_+$ satisfying $p \equiv q \equiv 3 \mod 4$; such numbers are referred to as Blum primes. Additionally, select an initial condition $1 < \sigma^0 \in \mathbb{N}_+$ that is coprime to m (i.e. $\nexists r \in \mathbb{N}_+ : \sigma^0 = rp \vee \sigma^0 = rq$). Then the output of this PRNG at time $k > 0$ is given by

$$\sigma^k = \left(\sigma^{k-1}\right)^2 \mod m. \tag{6}$$

The cryptographic security of the BBS algorithm relies on the quadratic residuosity problem, subject to various other conditions that are outside the scope of this paper. For more information the reader is referred to [18].

It is important to note that the origin is not in the codomain of the above generator as shown by the following lemma.

Lemma 1: Let $p, q \in \mathbb{N}_+$ be two distinct prime numbers. Then

$$\nexists z \in \mathbb{N}_+ : z < pq, z^2 \equiv 0 \mod pq. \tag{7}$$

Proof: First note that (7) can be restated as

$$\nexists z, \alpha \in \mathbb{N}_+ : z < pq, z^2 = \alpha pq. \tag{8}$$

Now suppose that (8) does not hold and let $\mathcal{A}$ and $\mathcal{Z}$ denote the prime factor multisets of α and z^2 respectively. Then the following statements must be true:

(i) $\{p, q\} \nsubseteq \mathcal{A}$.
(ii) $\mathcal{Z} = \mathcal{A} \cup \{p, q\}$.
(iii) Every element of $\mathcal{Z}$ has even multiplicity.

Since $z < pq$ then $z^2 < (pq)^2$ hence $\alpha < pq$ and (i) follows. Statements (ii) and (iii) follow trivially from $z^2 = \alpha pq$.

Combining statements (ii) and (iii) above suggests that both p and q must belong to $\mathcal{A}$ with odd multiplicity. This however violates statement (i) leading to a contradiction which concludes the proof. ■

The proposed method extends (6) by incorporating historical state information of the transmitting node. Let m and σ^0 satisfy the same conditions as above. Furthermore, suppose that x_i^k is the state vector to be transmitted, then a time varying hash key can be generated by

$$\sigma_i^k = \left(\sigma_i^{k-1} + \psi\left(x_i^{k-1}, \mathcal{L}\right)\right)^2 \mod m \tag{9}$$

where $\psi : \mathbb{R}^{n_i} \times \mathcal{L} \mapsto \mathbb{N}_+$ is a function with parameter set $\mathcal{L}$ which determines how the state measurements affect the generated key. Note that if ψ is chosen such that $\sigma_i^{k-1} + \psi\left(x_i^{k-1}, \mathcal{L}\right) < m \ \forall x_i^{k-1} \in \mathbb{R}^{n_i}$, then Lemma 1 can be invoked to prove that $\sigma_i^k \neq 0 \ \forall k \in \bar{\mathbb{N}}_+$. One possible selection that satisfies this condition is

$$\psi\left(x_i^{k-1}, \{\Lambda, \gamma\}\right) = \left\lfloor (m - \sigma_i^{k-1}) \frac{\|\Lambda x_i^{k-1}\|^2 + 1}{\|x_i^{k-1}\|^2 + \gamma} \right\rfloor \tag{10}$$

for some nonsingular matrix $\Lambda \in \mathbb{R}^{n_i \times n_i}$ satisfying $\|\Lambda\| < 1$ and $1 < \gamma \in \mathbb{R}$. The full rank restriction is imposed such that the case where $x_i^{k-1} \in \ker \Lambda$ is avoided, which would minimize the effect of x_i^{k-1} on σ_i^k, especially as $\|x_i^{k-1}\| \to \infty$. Using the Rayleigh quotient theorem [19, Theorem 4.2.2], it can be shown that the quotient in (10) admits values in the interval $\left[\min\left\{\underline{\lambda}(\Lambda^*\Lambda), \gamma^{-1}\right\}, \max\left\{\|\Lambda\|^2, \gamma^{-1}\right\}\right] \subseteq (0, 1)$ where $\underline{\lambda}(A)$ denotes the minimum eigenvalue of the matrix $A \in \mathbb{R}^{n \times n}$. The lower and upper bounds are achievable only if $\underline{\lambda}(\Lambda^*\Lambda) \geq \gamma^{-1}$ and $\|\Lambda\|^2 \leq \gamma^{-1}$ respectively. This then implies that $\psi\left(x_i^{k-1}, \{\Lambda, \gamma\}\right) < m - \sigma_i^{k-1} \ \forall x_i^{k-1} \in \mathbb{R}^{n_i}$, as desired. It is important that Λ is not chosen as $\Lambda = \gamma^{-\frac{1}{2}} U$ for some unitary matrix $U \in \mathbb{R}^{n_i \times n_i}$ since this leads to

$$\frac{\|\Lambda x_i^{k-1}\|^2 + 1}{\|x_i^k - 1\|^2 + \gamma} = \frac{\gamma^{-1}\|x_i^{k-1}\|^2 + 1}{\|x_i^{k-1}\|^2 + \gamma} = \frac{1}{\gamma}.$$

Hence (10) reduces to $\psi\left(x_i^{k-1}, \{\Lambda, \gamma\}\right) = \lfloor \gamma^{-1}(m - \sigma_i^{k-1}) \rfloor$ and x_i^{k-1} does not contribute to the updated hashing key.

The modification proposed by the combination of (9) and (10) offers several advantages. First, unlike the original BBS formulation or any other PRNG, the output of the algorithm at time k cannot be directly evaluated from the initial conditions without explicit, complete and detailed knowledge of all transmitted measurements x_i^l for all $l \in [0, k-1]$. Secondly, the minute variations (due to measurement noise, disturbances, etc.) in x_i^k increase entropy and therefore the security of the generated key. Finally, periodic behavior is no longer an issue; in contrast to other PRNGs (including BBS) where periods are guaranteed to exist in the output sequence.

Remark 1: As previously stated, ϕ_i and S_i are not unique. For example, the quotient in (10) could be replaced by the logistic function, hyperbolic tangent, or any other sigmoid function while maintaining the desired properties. ♦

Remark 2: The result presented in Lemma 1 can be strengthened to state that for any $2 \leq l \in \mathbb{N}_+$ and $z \in \bar{\mathbb{N}}_+$

$$z^l \equiv 0 \mod pq \Leftrightarrow z \equiv 0 \mod pq.$$

Then the allowable set for $\psi\left(x_i^{k-1}, \mathcal{L}\right)$ is considerably increased in size, with the only restriction being that

$$\sigma_i^{k-1} + \psi\left(x_i^{k-1}, \mathcal{L}\right) \not\equiv 0 \mod pq \quad \forall x_i^{k-1} \in \mathbb{R}^{n_i}.$$

Furthermore, this implies that (9) could be modified to exploit the intractability of the l^{th}-residuosity problem thus further improving the security of the proposed scheme. ♦

V. Numerical Example

The numerical example presented in section III will now be reconsidered with the proposed method implemented using the SHA-1 hash function. As there are no known feasible collision attack strategies against HMAC [20], it is hard to construct a realistic attack model for this example. It is assumed that the attacker simply substitutes the relevant state measurements at will while keeping the transmitted hash h_i^k unaltered. The injected data is identical to that used in Figure 4 generated according to (3a) and (3b).

Once the attack is detected by P_2 via a hash mismatch, it requests its position relative to P_1 from P_3. Since the integrity and authenticity of the data received by P_3 is validated using the hash value, it can be trusted and thus passed along to P_2. This imposes a unit delay in the control loop; such problems have been extensively studied in the past (see for example [21]). Clearly, if either of the edges $e_{1,3}$ or $e_{3,2}$ was also found to be compromised then the mission

would not be achievable as a trusted communication path from P_1 to P_2 would not be available.

Figure 6 shows the results of this simulation. It is noted that the effects of the attack are hard to observe by inspection of this figure when comparing to Figure 3. This is an indicator that the proposed method is functioning as expected.

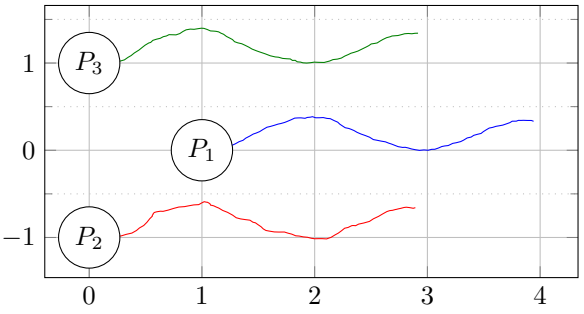
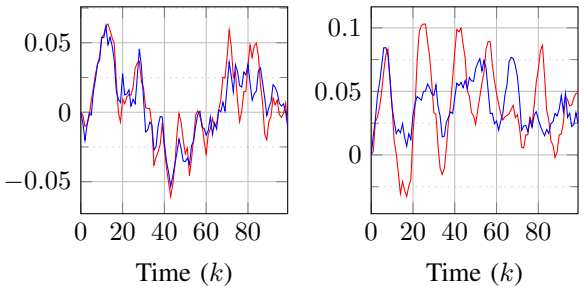


Fig. 6: Robot formation under attack using the proposed method ($x-y$ position plot).



(a) x position tracking error (b) y position tracking error

Fig. 7: Relative position errors $x_1^k - x_2^k + r_{1,2}$ (——) and $x_1^k - x_3^k + r_{1,3}$ (——), where $r_{1,2}$ and $r_{1,3}$ denote the desired relative position vectors and are given by $r_{1,2} = -[1\ 1]^T$ and $r_{1,3} = [1\ -1]^T$.

The effect of the attack on the performance of the closed loop system is easier to observe in the relative position errors between the leader and the two followers; plotted in Figure 7. It can be seen that following the attack at $k = 10$, the performance of the control system stabilizing $x_1^k - x_2^k$ deteriorates; however, stability is maintained and the desired trajectory is successfully tracked.

VI. Conclusion

A simple method has been proposed to detect false data injection attacks in networks of cyber-physical systems where state measurements are communicated between vertices. This is inherently decentralized and relies on well established cryptographic tools. No information about the plant being monitored or the structure of the communication graph is required. Although full state measurement is considered herein, the same technique could be used for any data set (measurement or otherwise) that must be transmitted.

The proposed method provides strong protection against this type of attack. The requirement of additional computational resources and transmission bandwidth, makes it unlikely to be used in established industrial control systems due to retrofitting costs. However, for high order systems, this approach would incur less computational overheads than observer based alternatives found in the literature. In fact, the complexity of this approach is $\mathcal{O}(n)$ (given the right choice of parameters) in contrast to an observer update the complexity of which is at best $\mathcal{O}(n^2)$. The developed technique would certainly be suitable for any newly commissioned plants as well as cases where the computational capacity already exists, such as the presented example of autonomous system formation control.

References

[1] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," *Survival*, vol. 53, pp. 23–40, 02 2011.

[2] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 ukraine blackout: Implications for false data injection attacks," *IEEE Transactions on Power Systems*, vol. 32, pp. 3317–3318, July 2017.

[3] F. Pasqualetti, F. Drfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, pp. 2715–2729, Nov 2013.

[4] S. Mishra, Y. Shoukry, N. Karamchandani, S. N. Diggavi, and P. Tabuada, "Secure state estimation against sensor attacks in the presence of noise," *IEEE Transactions on Control of Network Systems*, vol. 4, pp. 49–59, March 2017.

[5] J. Y. Keller and D. Sauter, "Monitoring of stealthy attack in networked control systems," in *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, pp. 462–467, Oct 2013.

[6] S. Ntalampiras, "Detection of integrity attacks in cyber-physical critical infrastructures using ensemble modeling," *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 104–111, Feb 2015.

[7] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *49th IEEE Conference on Decision and Control*, pp. 5967–5972, Dec 2010.

[8] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding schemes for securing cyber-physical systems against stealthy data injection attacks," *IEEE Transactions on Control of Network Systems*, vol. 4, pp. 106–117, March 2017.

[9] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, pp. 13:1–13:33, June 2011.

[10] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic Control*, vol. 59, pp. 1454–1467, June 2014.

[11] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, pp. 4203–4215, Dec 2005.

[12] C. De Persis and P. Tesi, "Input-to-state stabilizing control under denial-of-service," *IEEE Transactions on Automatic Control*, vol. 60, pp. 2930–2944, Nov 2015.

[13] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," RFC 2104, RFC Editor, February 1997.

[14] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, Inc., 1st ed., 1996.

[15] G. Chartrand and P. Zhang, *A First Course in Graph Theory*. Dover books on mathematics, Dover Publications, 2012.

[16] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, p. 644654, 1976.

[17] T. E. Hull and A. R. Dobell, "Random number generators," *SIAM Review*, vol. 4, no. 3, pp. 230–254, 1962.

[18] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on Computing*, vol. 15, no. 2, p. 364383, 1986.

[19] R. Horn and C. Johnson, *Topics in Matrix Analysis*. Topics in Matrix Analysis, Cambridge University Press, 1994.

[20] S. Turner and L. Chen, "Updated security considerations for the MD5 message-digest and the HMAC-MD5 algorithms," RFC 6151, RFC Editor, March 2011.

[21] K. Gu and S.-I. Niculescu, "Survey on recent results in the stability and control of time-delay systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 125, no. 2, pp. 158–165, 2003.