Towards Trustworthy Data in Networked Control Systems: A Hardware-Based Approach

Thomas Ulz, Thomas Pieber, Christian Steger
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
{thomas.ulz, thomas.pieber, steger}@tugraz.at

Rainer Matischek, Holger Bock
Development Center Graz
Infineon Technologies Austria AG
Graz, Austria
{rainer.matischek, holger.bock}@infineon.com

***Abstract*—The importance of Networked Control Systems (NCS) is steadily increasing due to recent trends such as smart factories. Correct functionality of such NCS needs to be protected as malfunctioning systems could have severe consequences for the controlled process or even threaten human lives. However, with the increase in NCS, also attacks targeting these systems are becoming more frequent. To mitigate attacks that utilize captured sensor data in an NCS, transferred data needs to be protected. While using well-known methods such as Transport Layer Security (TLS) might be suitable to protect the data, resource constraint devices such as sensors often are not powerful enough to perform the necessary cryptographic operations. Also, as we will show in this paper, applying simple encryption in an NCS may enable easy Denial-of-Service (DoS) attacks by attacking single bits of the encrypted data. Therefore, in this paper, we present a hardware-based approach that enables sensors to perform the necessary encryption while being robust against (injected) bit failures.**

***Index Terms*—Networked Control System; Security; Encryption; Forward Error Correction.**

I. Introduction

Networked Control System (NCS) nowadays are gaining popularity due to, among other things, Internet of Things (IoT) technologies where systems such as intelligent traffic control systems comprising of a large number of sensors and actuators are envisioned [1]. Systems that monitor and control a physical process through some computational device are often generally defined as cyber-physical systems (CPS) [2]. These systems also allow the involved devices to be connected to private and even public networks. Inspired by the IoT and CPS, several working groups proposed high-tech strategies such as Industry 4.0 [3] or smart manufacturing [4]. These strategies envision so-called smart factories that connect every device involved in the production process with each other or even with the Internet. All of these trends have one thing in common: devices are interconnected which allows NCS to be implemented efficiently using the corresponding network structures.

A general definition for an NCS is given by Gupta and Chow [5]. The authors state that a traditional feedback control system that is closed via a shared communication channel should be classified as an NCS. They also highlight this as a key characteristic common in many NCS definitions: information in the NCS is exchanged between involved components (sensor, controller, and actuator) using this shared communication channel. However, using a shared communication channel results in several challenges for NCS:

1) **Delays:** Using a shared communication channel may induce unreliable and non-deterministic behaviour into an NCS [6]. If the resulting delays are too large for an NCS with time constraints, the performance of the NCS can be impacted [7]. This could ultimately lead to potential physical damage to the controlled process or even threaten human lives, for example, in traffic NCS.
2) **Packet Loss:** Another property common in shared communication channels is the probability of packet loss. If relevant information such as measured plant output or control input are lost, the stability of the NCS may be compromised [8]. Stabilization problems could lead to compromised NCS performance, severe physical damage of the controlled process, or even threaten lives.
3) **Information Security:** When transferring information such as measured output or control input using a shared communication channel, attacks that could compromise the NCS functionality can easily be conducted [9]. In addition to that, an adversary that has learned the behaviour of an NCS through eavesdropping communication, may be able to manipulate a system in a way such that the attack remains undetected [10]. Therefore, the trustworthiness of transferred information often needs to be improved.

While a lot of current research is dedicated to the impact of network delays and packet loss in NCS, not much research has been done regarding information security as pointed out by Byres and Lowe [11]. One of the limiting factors in NCS related security research is the fact that security measures require additional computational resources and time. For example, using TLS for sensor to controller communication often will be infeasible due to resource constraint sensor hardware. However, the trustworthiness of sensor data is essential in NCS as compromised data can lead to malfunctioning systems. To improve the trustworthiness of data while imposing as little delay as possible, algorithms and/or hardware extensions will

978-1-5090-6505-9/17/$31.00

be necessary. The approach presented in this paper therefore makes the following contributions: (i) We propose the combination of encryption and error correction to mitigate NCS related attacks. (ii) To impose a minimum of delay, a hardware extension is presented that can be integrated into sensors and actuators. (iii) The presented approach can be applied to the general concept of NCS; no network technology and related security feature such as error correction is assumed.

The remainder of this paper is structured as follows. In Section II technical background information regarding technologies used in our approach as well as related work regarding NCS security solutions is given. Section IV and V show a naïve and an enhanced approach respectively that can be used to mitigate certain kinds of attacks. This paper is then concluded in Section VI where also potential future work is discussed.

II. Background and Related Work

A. Authenticated Encryption (AE)

AE generally combines a symmetric encryption scheme with a message authentication code (MAC) to provide confidentiality, integrity and authenticity of data [12]. A symmetric (private-key) encryption scheme requires the two communicating parties to be in possession of the same shared secret key [13]. A widely used symmetric encryption scheme, the advanced encryption standard (AES) [14], operates as a block cipher, processing plaintext blocks of 16 bytes. For most symmetric block ciphers specialized AE modes exist, such as AES-CCM that can efficiently be implemented in hardware [15] to provide reliable and fast execution of the algorithm. Such implementations are feasible to provide encryption fast enough for 100 Gbit/s ethernets [16].

B. Forward Error Correction (FEC)

FEC is used to detect and correct errors in data transmission resulting from unreliable and noisy communication channels. First applied by Hamming [17], the basic idea is to add redundant information produced by an error correcting code (ECC) before sending data. This redundant information allows to detect or probably even to correct errors without requiring the data to be transmitted again. ECCs, however, are not limited to data transmission as one major field of application is memory [18]. FEC can be implemented efficiently enough to be suitable for applications relying on high speed, such as 100 Gbit/s transport networks [19]. A high performance type of FEC are so-called turbo codes [20] that are used in 3G and 4G networks as well as in space programs [21].

C. Joint Encryption and Error Correction (JEEC)

JEEC was already discussed in research in the 1980s where authors claimed that combining encryption and error correction could lead to efficient implementations that could be done in a cost effective way [22], [23]. These solutions used the data encryption standard (DES) that nowadays is ousted by AES. Mathur et al. [24] present an approach based on AES that provides the same security level as AES. Gligoroski et al. [25] discuss encryption and error correction coding done in a single step for more recent algorithms. As the authors mention, also sequential execution of encryption and error correction codes is a possibility with execution performance being a drawback of that approach.

D. Security Controller (SC)

SCs are processing units that provide a secured execution environment for applications as well as secured storage for data and applications. Compared to a general purpose CPU, attacks based on issues such as buffer overflows are much harder to exploit on an SC. In addition, SCs also provide tamper resistance [26] that mitigates physical attacks by using appropriate countermeasures. To assess the provided security level of an SC the common criteria (CC) information technology security evaluation is used [27]. Because embedded systems are often operated in untrusted environments and thus accessible to adversaries, tamper resistance is of critical importance [28].

E. NCS Security

In order to understand security threats in NCS, a threat model containing potential vulnerabilities and the impact of attacks needs to be defined first, as shown by Cárdenas et al. [29]. The authors also highlight the differences of NCS compared to traditional IT components: (i) frequent security updates may not be possible for NCS and (ii) the interaction of NCS with the physical world that greatly increases the impact of attacks. For an NCS to be considered secured, the following four security properties need to be fulfilled:

- **Confidentiality**: Data is not made available to unauthorized entities.
- **Integrity**: Data is not modified in an undetected manner during its entire life cycle.
- **Availability**: System is available in order to fulfill its intended task.
- **Authenticity**: Data is from the expected sender and not injected by some other entity.

These properties can be compromised by different types of NCS related attacks. Cárdenas et al. [9] highlight five attacking points for CPS (see Fig. 1). Due to the similarities between CPS and NCS, all of these five attacking points also apply for NCS. Attacks of category **A1** directly target the physical process. **A2** attacks are so-called *deception* attacks that are characterized by adversaries inducing false information $\tilde{y} \neq y$. The attacks can be backed by a previous *learning phase* in which the expected behaviour of the plant is learned first [10]. **A3** represents Denial-of-Service (DoS) attacks on the sensor to controller communication channel. Attacks that are characterized by adversaries trying to induce false control commands $\tilde{u} \neq u$ are represented by **A4**. Here, the adversary could either target the controller or the communication channel. **A5** denotes DoS attacks on the controller to actuator channel.

DoS attacks (**A3, A5**) are well covered in research with many authors trying to account for these types of attacks in the controller [30]. Due to the networking nature of NCS,

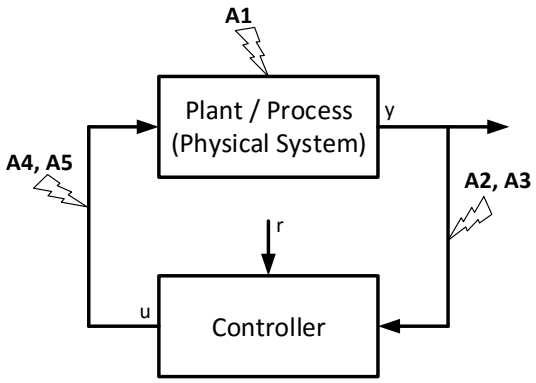


Fig. 1: Potential attacking points in a CPS (adapted from Cárdenas et al. [9]).

TABLE I: Comparison with related work. The properties (C)onfidentiality, (I)ntegrity, (A)vailability, and (Au)thenticity are evaluated.

Related Work	Remark	C	I	A	Au
[32], [30]	Packet loss due to DoS attacks.	✗	✗	✓	✗
[33], [10]	Deception attack detection.	✗	✗	✓	✗
[34]	Encryption and hash algorithms applied; algorithms considered insecure.	✓	✓	✗	✗
[35]	Encryption applied.	✓	✗	✗	✗
Our approach	Suitable combination of algorithms; tamper resistant hardware.	✓	✓	✓	✓

unintentional packet loss is a characteristic that robust control algorithms need to account for [31]. The approaches used to model such unintentional packet loss can then be adapted to account for malicious packet jamming or compromising. Amin et al. [32] use optimal control theory tools to optimize controller performance such that safety specifications are satisfied with high probability while power limitations are considered.

Replay attacks or deception attacks (**A2**) target sensor data in an NCS to learn the expected behaviour of a plant and then use that data to inject false measurements. These false measurements can be used to hide an ongoing attack or to compromise the functionality of an NCS. Mo and Sinopoli [33] discuss the impact of such attacks that can target a system in its steady state. They also propose a method to detect an ongoing replay attack that however decreases the performance of their used controller algorithm. Mo et al. [36] also demonstrate the usage of injected false sensor data to compromise the functionality of a state estimator, thus directly targeting the functionality of an NCS.

Urbina et al. [10] discuss the impact of stealthy deception attacks on control system. They suggest to use a physics-based attack detection model to detect ongoing attacks. The main idea of their approach is to compare current properties of the system with physics-based model of the system under normal behaviour. Pang and Liu [34] propose to use data encryption standard (DES) encryption and MD5 hashes to increase the confidentiality, integrity, and authenticity of transferred packets in an NCS. However, the approach by the authors has three problems: (i) Both used algorithms, DES and MD5 are considered to be insecure nowadays [37], [38]. (ii) Plain hash functions such as MD5 can not be used to efficiently protect message authenticity [39]. (iii) All security measures are implemented in software by the authors. This increases delays as well as allows keys to be extracted by physical attacks [40]. Gupta and Chow [35] analyze additional delay in NCS induced by security algorithms such as DES, 3DES, and AES. In the experiment conducted by the authors only DES encryption is considered as fast enough to not compromise the stability of the NCS. To compensate the overhead for other algorithms, the authors suggest to use 1-D gain schedulers.

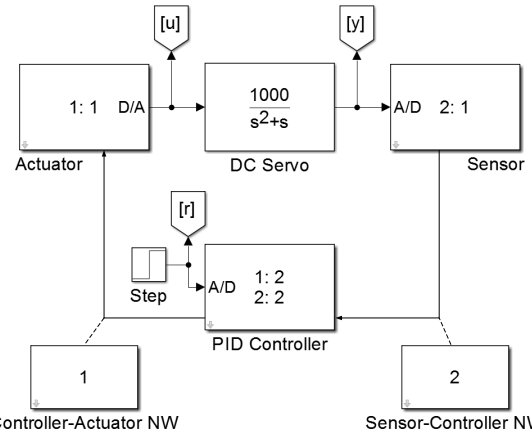


Fig. 2: MATLAB/Simulink model of an NCS for a DC servo. Both involved network connections are additionally outlined.

To increase the trustworthiness of data in an NCS, tradeoffs between measures such as imposed delay, provided security level, or energy efficiency need to be made [41] due to constraint devices. However, these tradeoffs might compromise security. In contrast to that, the approach presented in this paper tries to keep the associated impact of including security such as delay as small as possible. A comparison of our approach with presented work is given in Table I.

III. Evaluation Environment

To demonstrate the impact of different measures and parameters applied to the NCS, we use a MATLAB/Simulink simulation [42]. In addition to that, the TrueTime toolbox [43] is used to simulate network related behaviour, scheduling of software components, and real-time aspects. The process used for evaluation in this paper is described by the transfer function given in (1).

$$G(s) = \frac{1000}{s(s+1)} \tag{1}$$

The system corresponding to that transfer function is a simple DC servo motor; the measurable system output being the angular position of that DC servo. To control this DC

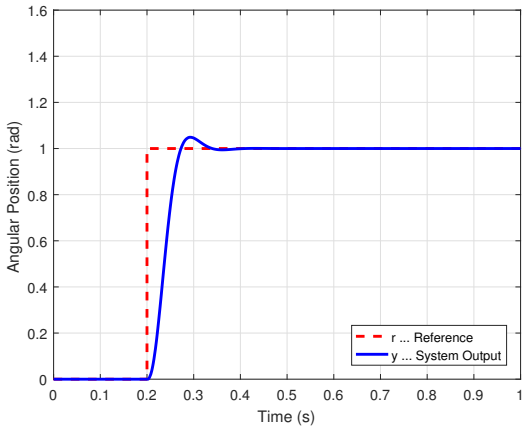


Fig. 3: Step response of the system shown in Fig. 2.

servo, a PD controller is used. Both the plant and the used PD controller can be found in the examples included in the TrueTime toolbox. The NCS shown in Fig. 2 comprises the DC servo plant, the respective actuators and sensors, the PD controller, and the simulated networks necessary for communication between components. As can be seen in Fig. 2, we use two different networks. Network 1 handles communication between the PD controller and the actuator, while the communication between the sensor and PD controller is handled by network 2. The PD controller, therefore is connected to both networks and has the same node ID 2 in all two networks. The actuator and sensor are assigned node ID 1 in their respective network. This setup allows us to simulate the impact of applied measures and network parameters on different parts of the NCS. In our case, we only manipulate the communication between sensor and PD controller. As imposed time delays are not a focus of this publication, the delays imposed by the network technology applied in an NCS are set low enough (10 ms for each transmission) for the simple PD controller to work correctly. The resulting closed-loop step response of our simulated NCS is shown in Fig. 3.

IV. Naïve Approach

The easiest approach to increase confidentiality, integrity, and authenticity of information transferred in an NCS is to use appropriate encryption algorithms. However, as we will show, this approach is also naïve in some kind as it introduces drawbacks regarding the NCS functionality that were to the best knowledge of the authors not discussed in other publications.

A. Usage of AE

In contrast to the discussed related work, we propose to use AE in the presented NCS context as this combination of encryption and MAC is suitable to provide confidentiality, integrity, and authenticity of information. Using encryption only or a combination of encryption and hash algorithms [34], [35] can not be used to provide all three mentioned security properties. If plain encryption is used, it is sufficient for an adversary to change a single bit of each transmitted packet to completely disturb the NCS functionality. As an example we use four sensor measurements shown in Table II as plaintext (PT) and encrypt them in one block using AES. One bit of the corresponding cyphertext (CT) is modified (last bit changed from 1 to 0) which results in a corrupted cyphertext (CT'). If this corrupted cyphertext is decrypted using the same key as for encrypting PT, a corrupted plaintext (PT') results. As can be seen, by just flipping one bit of the cypertext, the plaintext does not correlate to the original sensor measurements in any way and thus, can cause severe problems in an NCS if this corrupt data is not detected.

TABLE II: Sensor data plaintext (PT), cyphertext (CT), and corrupted cyphertext (CT') with resulting plaintext (PT').

	Sensor 1	**Sensor 2**	**Sensor 3**	**Sensor 4**
PT	0x00000001	0x00000002	0x00000003	0x00000004
CT	0xDE154CCE	0x18E65A6E	0xBD9A0593	0xE1B82507
CT'	0xDE154CCE	0x18E65A6E	0xBD9A0593	0xE1B82506
PT'	0x2D3DB30D	0xE89541F5	0x9AFD9AED	0x03BD8985

TABLE III: AES modes for AE and the corresponding performance measures from Crypto++ [44].

Algorithm	**MiB/Second**	**Cycles per Byte**	**Table**
AES GCM 2K	102	17.2	2K
AES GCM 64K	108	16.1	64K
AES CCM	61	28.6	-

To detect problems resulting from manipulated cyphertexts and to provide data confidentiality, integrity, and authenticity we propose to use AE. AE can be implemented by combining encryption with a MAC. AES modes that can be used for AE are Counter with CBC-MAC Mode (CCM) as well as Galois/Counter Mode (GCM) [14]. We propose to choose the corresponding mode based on the memory/execution time tradeoff that needs to be made between those two algorithms as shown by Crypto++ Benachmarks [44] in Table III. If execution time is the most relevant factor, AES with GCM should be used for AE.

B. Bit Failures and Block ciphers

If AE based on a block cipher is applied, malicious data packages can be detected and discarted. However, this property is problematic for NCS as a single flipped bit causes a package containing sensor data to be dropped. For example, multi user Ethernet has a typical bit error rate (BER) of about 10^{-9} [45]. The packet error rate (PER) can be calculated according to (2) where N is the packet's size in bits.

$$PER = 1 - (1 - BER)^N \tag{2}$$

For transmitting 1 kB of data (N=8000) this equates to a PER of $\approx 8{\cdot}10^{-6}$. However, if an adversary is able to inject bit

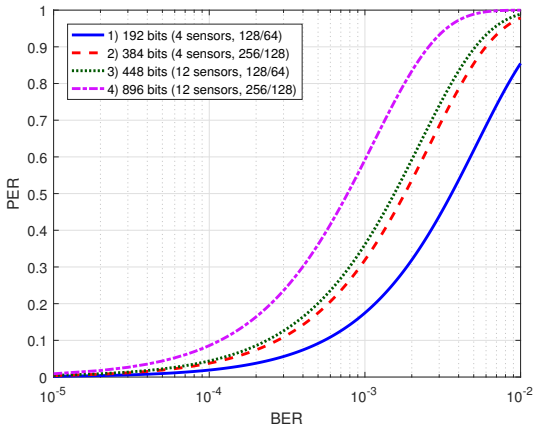


Fig. 4: PER depending on the BER, payload size, and parameters of the applied cryptographic algorithms.

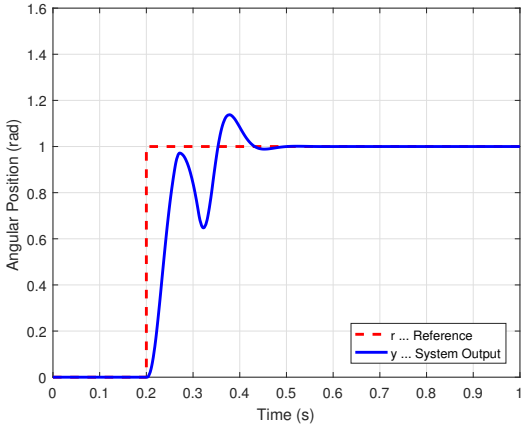
failures in any way, for example, by jamming wireless signals, the PER increases rapidly as can be seen in Fig. 4. In this figure, the PER depending on the BER, the payload size and the parameters of the applied cryptographic functions is shown. We demonstrate four different scenarios with combinations of AES block size, MAC length and number of sensors there. We assumed that each sensor measurement can be represented by a 32 bit number in this example.

1) 192 bits payload: AES block size 128 bit, 4 sensor measurements: 1 AES block, 64 bit MAC
2) 384 bits payload: AES block size 256 bit, 4 sensor measurements, 1 AES block, 128 bit MAC
3) 448 bits payload: AES block size 128 bit, 12 sensor measurements, 3 AES blocks, 64 bit MAC
4) 896 bits payload: AES block size 256 bit, 12 sensor measurements, 3 AES blocks, 128 bit MAC

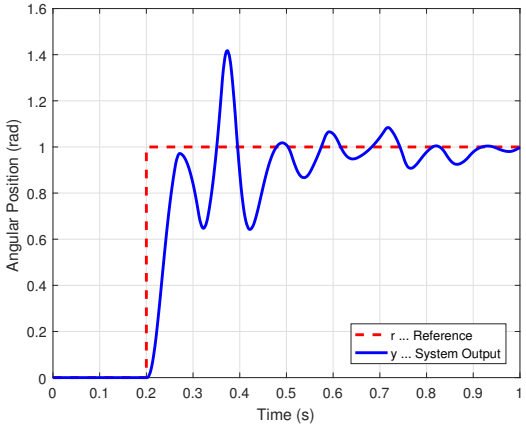
An adversary that is able to manipulate just one in 1000 transmitted bits, causes a PER between 20% and 60% in our examples (Fig. 4). To highlight the impact of such a high PER, the system presented in Section III is simulated again with 25% packet loss and 50% packet loss (between sensor and controller) respectively. These simulations result in the step responses shown in Fig. 5. For 25% packet loss the given reference input can be achieved by the system (Fig. 5a) although it takes longer to reach the desired reference value when compared to the standard case shown in Fig. 3. When simulating 50% of packet loss between sensors and controller (Fig. 5b) the given reference input is hardly reached by the system. Thus, by trying to prevent deception attacks, simply applying encryption might make DoS attacks a lot easier for adversaries.

C. Stream Ciphers

One potential technology to mitigate the problems related to bit failures in cyphertexts are stream ciphers [46]. Stream ciphers encrypt each plaintext bit separately by combining it in some specified form with a corresponding bit of a keystream.



(a) Step response of system with 25% packet loss.



(b) Step response of system with 50% packet loss.

Fig. 5: Step responses for the system shown in Fig. 2 with different amounts of packet loss in network 2.

Due to this property, stream ciphers are not prone to the previously described problem; flipping one bit in the cyphertext results in one corrupted bit in the plaintext after decryption. However, the most widely used stream cipher Rivest Cipher 4 (RC4) is considered insecure due to various vulnerabilities [47] and was therefore removed from TLS [48]. Other stream ciphers, such as Salsa20 [49] are not yet widely proven to be considered. However, future developments regarding stream ciphers need to be monitored regarding their potential impact on NCS security.

As we have shown, simply applying encryption to prevent deception attacks in NCS is not an applicable approach. Therefore, in the next section, we present an enhanced approach that mitigates the drawbacks of using encryption in NCS.

V. Enhanced Approach

As simply applying encryption in an NCS has drawbacks regarding bit failures, we propose an enhanced approach that combines encryption and error correction to mitigate these drawbacks. Depending on the used network technology in

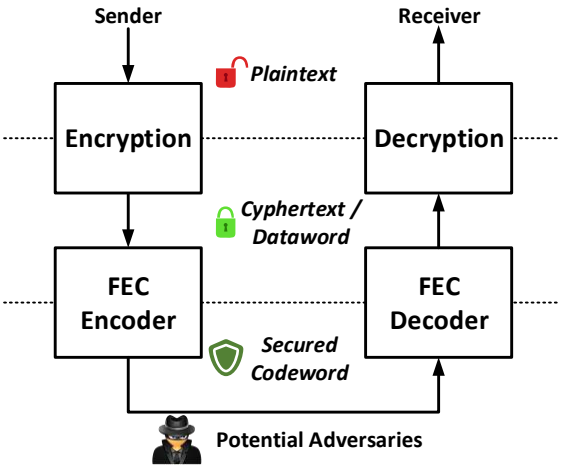


Fig. 6: Sequential JEEC applied in our approach.

an NCS, either encryption, error correction, or both technologies are applied, for instance when using TLS over an Ethernet channel. However, to make NCS security measures independent of the applied network technologies, we propose to include our approach directly into the involved components (sensors, actuators, controller).

A. JEEC

In our approach, we sequentially combine encryption and error correction to reduce the impact of bit failures in cyphertexts. More specifically, we suggest to use a suitable AES mode for AE (GCM or CCM) in combination with turbo codes FEC. We suggest this combination of algorithms for the following reasons:

- AES is a well established symmetric algorithm.
- AES provides modes of operation to apply AE.
- Turbo codes are very fast FEC algorithms that can be implemented efficiently in hardware.
- The performance of turbo codes regarding the achievable BER is close to the Shannon limit [50].

Rao [23] states that the sequential order of encryption and error correction is irrelevant, as long as both are performed. This, however is not true for our proposed approach. If the error correction would be executed before encrypting the complete data package, no advantage compared to simply applying encryption could be achieved. Therefore, we propose to first encrypt the plaintext, followed by performing the FEC encoding before sending data as shown in Fig. 6.

By using this approach, a plaintext is encrypted and the resulting cyphertext is seen as the dataword that is used as input for the FEC encoding. The resulting codeword then resembles the previous cyphertext plus data redundancy that was added by the FEC encoding. On the receiving end, the data first needs to be decoded before decrypting the corresponding cyphertext. This process allows the transmitted data to be protected by AE as well as by FEC in order to provide confidentiality, integrity, and authenticity of data and to make DoS attacks harder compared to simply applying encryption.

B. Analysis of Functionality

In contrast to our sequential approach there are also approaches that combine encryption and error correction in a single step [24], [25]. However, we propose to use separate encryption and error correction algorithms as the provided functionalities are easier to verify and proof for both components respectively. Moreover, this separation of components allows the security relevant parts to be executed on dedicated hardware to increase the provided level of security.

The security properties of AE based on suitable AES modes are demonstrated in literature [12], [14]. AES is the most used symmetric encryption algorithm, and no severe weaknesseses in the algorithm were known at the time this publication was written. The functionality of the proposed turbo codes FEC is measured in the improvement in BER compared to using no FEC. For a fixed "signal to noise" ratio (E_b/N_0), a channel using turbo codes FEC provides a BER that is lower by a factor of 10^4 compared to an unencoded channel [51]. Thus, reducing the impact of (malicious) bit errors when transmitting data in an NCS.

C. Anomaly Detection

In addition to mitigating problems related to bit failures, also anomly detection [52] could be performed using the applied FEC. A very simple approach would be to define a threshold above the expected BER of a communication channel. If the encountered bit errors are then monitored in a certain time window and exceed this specified threshold, an anomaly could be reported. However, more complex mechanisms can be implemented based on this information. We will consider such mechanisms for future work.

D. Hardware Enhancements

Due to the real-time aspects of NCS coupled with often resource constraint devices, we also propose to include dedicated hardware components into sensors, actuators, and controllers in order to provide reliable execution times. In addition to that, dedicated hardware also provides additional security features that we are going to discuss in this section. Fig. 7 illustrates an NCS with included additional hardware components necessary for our presented approach. In this figure, SC denotes a so-called security controller, while EN and DE are FEC encoders and decoders respectively.

To allow security enhancing components to be included easily into sensors, actuators, and controllers, we propose a so-called JEEC enhancement. Due to its included interfaces, the JEEC enhancement shown in Fig. 8 can easily be integrated into NCS components. The JEEC enhancement consists of the following three components:

1) **CPU:** The general purpose CPU offers interfaces to sensors and actuators as well as to the communication channel. All necessary computations such as data preprocessing or the network stacks are handled by this CPU. In addition, the CPU needs to have interfaces to the SC and FEC components.

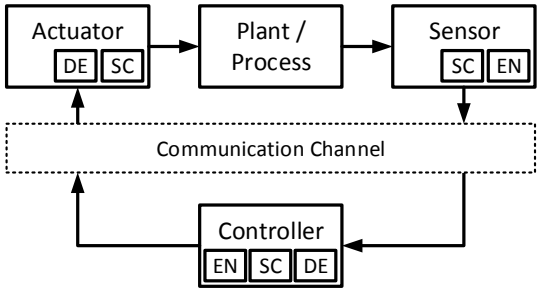


Fig. 7: Block diagram of NCS secured with our presented approach. In this block diagram, EN denotes a FEC encoder while DE denotes a FEC decoder.

2) **SC:** The SC is used to perform cryptographic operations in a secured environment. In addition to that, the SC also provides secured storage for confidential information such as key material. To provide these two functionalities, the SC needs to provide tamper resistance in order to mitigate physical attacks that try to extract or reveal confidential information. In addition, the SC also provides protection against software based attacks. A product line of SCs suitable for smart factories is, for example, offered by Infineon [53].
3) **FEC:** The FEC component is responsible to perform FEC calculations as efficient as possible. Due to constraints in size and/or price of sensors and actuators, the FEC component can also be split into decoder and encoder depending on the specified requirements.

Due to the necessary network functionality of components in an NCS, a network interface needs to be included in any case. Most of the time this also requires the inclusion of a CPU to handle the resulting overhead. Therefore, the proposed JEEC enhancement only requires to add an SC and the FEC component in most cases.

E. Advantages of Approach

The presented approach of using JEEC supported by dedicated hardware components has five advantages compared to current state of the art approaches: (i) AE provides confidentiality, integrity, and authenticity of data in a NCS; therefore, deception attacks can be mitigated. (ii) The combination of AE with FEC helps to mitigate the drawbacks resulting from bit failures in the transferred cyphertext. Despite bit failures, the same step response as shown in Fig. 3 can be achieved. Due to the sequential execution of encryption and error correction, the functionality of both components is not compromised. (iii) The information obtained in the error correction process can be used to perform additional anomaly detection. (iv) The presented JEEC enhancement can easily be included in any NCS component and thus increase the security of transferred data in an NCS. Due to using dedicated hardware components, constant runtime of cryptographic algorithms can be provided. (v) The tamper resistance provided by the SC can be used to protect confidential data if, for example, sensors are deployed where they are accessible by potential adversaries.

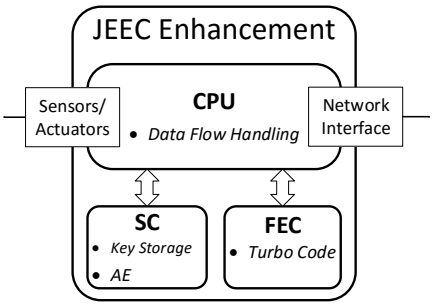


Fig. 8: JEEC enhancement for components in an NCS.

VI. Conclusion and Future Work

In this paper we have shown an encryption only approach to mitigate deception attacks in NCS. We highlight drawbacks of simply applying encryption and show the potential impact of (injected) bit failures in a NCS. To counteract these drawbacks, we propose to use JEEC to protect data confidentiality, integrity, and authenticity while also limiting the impact of adversaries that are able to artificially increase the BER in the used communication channel. We also presented a JEEC enhancement that can easily be integrated into NCS components while providing increased security and keeping delays as low as possible. As future work we plan to further investigate stream ciphers and JEEC algorithms that are able to perform encryption and error correction in a single step which might provide additional advantages.

Acknowledgment

This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 692480. This Joint Undertaking receives support from the European Unions Horizon 2020 research and innovation programme and Germany, Netherlands, Spain, Austria, Belgium, Slovakia.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
[2] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-Physical Systems: The Next Computing Revolution," in *Proceedings of the 47th Design Automation Conference*. ACM, 2010, pp. 731–736.
[3] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, *Recommendations for Implementing the strategic initiative INDUSTRIE 4.0: securing the future of German manufacturing industry; final report of the Industrie 4.0 working group*. Forschungsunion, 2013.
[4] J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, "Smart manufacturing, manufacturing intelligence and demand-dynamic performance," *Computers & Chemical Engineering*, vol. 47, pp. 145–156, 2012.
[5] R. A. Gupta and M.-Y. Chow, "Networked Control System: Overview and Research Trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.
[6] S. Soucek and T. Sauter, "Quality of Service Concerns in IP-Based Control Systems," *IEEE transactions on Industrial Electronics*, vol. 51, no. 6, pp. 1249–1258, 2004.
[7] G. Xie and L. Wang, "Stabilization of Networked Control Systems with Time-Varying Network-Induced Delay," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 4. IEEE, 2004, pp. 3551–3556.
[8] J. Xiong and J. Lam, "Stabilization of linear systems over networks with bounded packet loss," *Automatica*, vol. 43, no. 1, pp. 80–87, 2007.

[9] A. A. Cárdenas, S. Amin, and S. Sastry, "Secure Control: Towards Survivable Cyber-Physical Systems," in *Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on*. IEEE, 2008, pp. 495–500.

[10] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the Impact of Stealthy Attacks on Industrial Control Systems," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1092–1105.

[11] E. Byres and J. Lowe, "The Myths and Facts behind Cyber Security Risks for Industrial Control Systems," in *Proceedings of the VDE Kongress*, vol. 116, 2004, pp. 213–218.

[12] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2000, pp. 531–545.

[13] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption," in *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*. IEEE, 1997, pp. 394–403.

[14] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer Science & Business Media, 2013.

[15] E. López-Trejo, F. Rodríguez-Henríquez, and A. Díaz-Pérez, "An FPGA Implementation of CCM Mode Using AES," in *International Conference on Information Security and Cryptology*. Springer, 2005, pp. 322–334.

[16] L. Henzen and W. Fichtner, "FPGA Parallel-Pipelined AES-GCM Core for 100G Ethernet Applications," in *ESSCIRC, 2010 Proceedings of the*. IEEE, 2010, pp. 202–205.

[17] R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[18] C.-L. Chen and M. Hsiao, "Error-Correcting Codes for Semiconductor Memory Applications: A State-of-the-Art Review," *IBM Journal of Research and Development*, vol. 28, no. 2, pp. 124–134, 1984.

[19] F. Chang, K. Onohara, and T. Mizuochi, "Forward Error Correction for 100 G Transport Networks," *IEEE Communications Magazine*, vol. 48, no. 3, pp. S48–S55, 2010.

[20] C. Berrou and A. Glavieux, "Turbo Codes," *Encyclopedia of Telecommunications*, 2003.

[21] K. S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones, and F. Pollara, "The Development of Turbo and LDPC Codes for Deep-Space Applications," *Proceedings of the IEEE*, vol. 95, no. 11, pp. 2142–2156, 2007.

[22] S. C. Kak, "Joint Encryption and Error-Correction Coding," in *1983 IEEE Symposium on Security and Privacy*, 1983.

[23] T. R. N. Rao, "Joint Encryption and Error Correction Schemes," in *ACM SIGARCH Computer Architecture News*, vol. 12, no. 3. ACM, 1984, pp. 240–241.

[24] C. N. Mathur, K. Narayan, and K. Subbalakshmi, "High Diffusion Cipher: Encryption and Error Correction in a Single Cryptographic Primitive," in *International Conference on Applied Cryptography and Network Security*. Springer, 2006, pp. 309–324.

[25] D. Gligoroski, S. J. Knapskog, and S. Andova, "Cryptcoding-Encryption and Error-Correction Coding in a Single Step," in *Security and Management*. Citeseer, 2006, pp. 145–151.

[26] R. Anderson and M. Kuhn, "Tamper Resistance - a Cautionary Note," in *Proceedings of the second Usenix workshop on electronic commerce*, vol. 2, 1996, pp. 1–11.

[27] D. Mellado, E. Fernández-Medina, and M. Piattini, "A common criteria based security requirements engineering process for the development of secure information systems," *Computer standards & interfaces*, vol. 29, no. 2, pp. 244–253, 2007.

[28] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper Resistance Mechanisms for Secure Embedded Systems," in *VLSI Design, 2004. Proceedings. 17th International Conference on*. IEEE, 2004, pp. 605–611.

[29] A. A. Cárdenas, S. Amin, and S. Sastry, "Research Challenges for the Security of Control Systems," in *HotSec*, 2008.

[30] A. Teixeira, H. Sandberg, and K. H. Johansson, "Networked Control Systems under Cyber Attacks with Applications to Power Networks," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 3690–3696.

[31] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of Control and Estimation Over Lossy Networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[32] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and Secure Networked Control Systems under Denial-of-Service Attacks," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2009, pp. 31–45.

[33] Y. Mo and B. Sinopoli, "Secure Control Against Replay Attacks," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 911–918.

[34] Z.-H. Pang and G.-P. Liu, "Design and Implementation of Secure Networked Predictive Control Systems Under Deception Attacks," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1334–1342, 2012.

[35] R. A. Gupta and M.-Y. Chow, "Performance Assessment and Compensation for Secure Networked Control Systems," in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*. IEEE, 2008, pp. 2929–2934.

[36] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False Data Injection Attacks against State Estimation in Wireless Sensor Networks," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 5967–5972.

[37] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*. Springer Science & Business Media, 2012.

[38] X. Wang and H. Yu, "How to Break MD5 and Other Hash Functions," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2005, pp. 19–35.

[39] M. Bellare, R. Canetti, and H. Krawczyk, "Message Authentication using Hash Functions: The HMAC Construction," *RSA Laboratories CryptoBytes*, vol. 2, no. 1, pp. 12–15, 1996.

[40] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Moderator-Ravi, "Security as a New Dimension in Embedded System Design," in *Proceedings of the 41st annual Design Automation Conference*. ACM, 2004, pp. 753–760.

[41] W. Zeng and M.-Y. Chow, "Optimal Tradeoff Between Performance and Security in Networked Control Systems Based on Coevolutionary Algorithms," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 3016–3025, 2012.

[42] B. Shahian and M. Hassul, *Computer-Aided Control System Design Using MATLAB*. Prentice Hall Professional Technical Reference, 1992.

[43] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzén, "How Does Control Timing Affect Performance? Analysis and Simulation of Timing using Jitterbug and TrueTime," *IEEE control systems*, vol. 23, no. 3, pp. 16–30, 2003.

[44] W. Dai, "Speed Comparison of Popular Crypto Algorithms," https://www.cryptopp.com/benchmarks.html, 3 2009, (Accessed on 01/27/2017).

[45] V. J. Hernandez, A. J. Mendez, C. V. Bennett, R. M. Gagliardi, and W. J. Lennon, "Bit-Error-Rate Analysis of a 16-User Gigabit Ethernet Optical-CDMA (O-CDMA) Technology Demonstrator Using Wavelength/Time Codes," *IEEE Photonics Technology Letters*, vol. 17, no. 12, pp. 2784–2786, 2005.

[46] T. W. Cusick, C. Ding, and A. R. Renvall, *Stream Ciphers and Number Theory*. Elsevier, 2004, vol. 66.

[47] A. Klein, "Attacks on the RC4 Stream Cipher," *Designs, Codes and Cryptography*, vol. 48, no. 3, pp. 269–286, 2008.

[48] A. Popov, "Prohibiting RC4 Cipher Suites," Internet Requests for Comments, RFC Editor, RFC 7465, February 2015. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7465.txt

[49] D. J. Bernstein, "The Salsa20 Family of Stream Ciphers," in *New stream cipher designs*. Springer, 2008, pp. 84–97.

[50] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. 1," in *Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2. IEEE, 1993, pp. 1064–1070.

[51] M. A. Jordan and R. A. Nichols, "The Effects of Channel Characteristics on Turbo Code Performance," in *Military Communications Conference, 1996. MILCOM'96, Conference Proceedings, IEEE*, vol. 1. IEEE, 1996, pp. 17–21.

[52] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks Against Process Control Systems: Risk Assessment, Detection, and Response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*. ACM, 2011, pp. 355–366.

[53] J. Haid, "Hardware-based solutions secure machine identities in smart factories," *Boards & Solutions*, pp. 10–13, 2016.