Realizing Greater System Robustness Through Combining CIP Safety™ and CIP Security™

Vivek Hajarnavis
Technology Manager
Rockwell Automation

Xiaobo Peng
Sr. Safety Architect
Rockwell Automation

Jack Visoky
Principal Engineer and Security Architect
Rockwell Automation

Steve Seidlitz
Sr. Project Engineer
Rockwell Automation

Presented at the ODVA
2022 Industry Conference & 21st Annual Meeting
March 9, 2022
San Diego, California, USA

Abstract

Users need industrial communication protocols supporting both safety and security, like CIP Safety and CIP Security. There have been concerns that these two technologies may interfere, especially that the addition of security may be detrimental to the Functional Safety Argument. In particular there have been concerns regarding the Bit Error Probability, BEP, to model in the communication system.

In this paper we have shown via analysis and Markov Model that the addition of CIP Security to CIP Safety improves the safety argument rather than interfering with it and that overly conservative bit error probabilities do not need to be assumed. This is mainly due to the collision resistant properties of the HMAC, the diffusion properties of encryption, and possible additional robustness via security mandated testing (e.g. testing mandated by the IEC 62443 V&V model).

Through the Markov model analysis, the device internal interface between the security layer (broadly black channel) and safety layer is identified as the most critical part. The security standard IEC 62443 has similar requirement to the systematic capability defined in the safety standard IEC 61508. Hence adding a security implementation to the black channel part of the device will decrease systematic errors and reinforce the interface part.

Keywords

Functional Safety, Cybersecurity, Robustness, Industrial Protocols, Industrial Ethernet, Risk Reduction, Risk Mitigation, Markov Model, IEC 62443, IEC 61508

Definition of terms

CIP:	Common Industrial Protocol
CPF:	Communication Profile Family
CRC:	Cyclic Redundancy Check
DTLS:	Datagram Transport Layer Security
E/E/PE:	Electrical/Electronic/Programmable Electronic
EMI:	Electromagnetic Interference
EUC	Equipment Under Control
FIT:	Failures in Time (1 FIT = 10^{-9} failures per hour)
FSCP:	Functional Safety Communication Profile
HMAC:	Hash-based Method Authentication Code
HW:	Hardware
IACS:	Industrial Automation Control Systems
IEC:	International Electrotechnical Commission
IT:	Information Technology
OT:	Operational Technology
PLC:	Programmable Logic Controller
SIL:	Safety Integrity Level
SCL:	Safety Communication Layer
SPDU:	Safety Protocol Data Unit
TLS:	Transport Layer Security

Introduction

The use of Ethernet-based communications in industrial applications is well established, with EtherNet/IP™ recognized as one of the leading protocols for achieving connectivity [1]. Since the EtherNet/IP specification was first published, it has gone through a process of evolution reflecting the needs of users of the network by adding additional capabilities. These include safety – a function that had historically been separated physically from primary control – and security, where productivity needs have driven increased OT-IT connectivity. This integration of functions that were separate historically has in turn highlighted the need for security measures to be implemented to provide assurances that neither the functionality of assets, nor the intellectual property within the operations of a facility are compromised. In addition, recognizing the desire to ensure that security breaches do not have an impact on the safety of people who are using machinery, the IEC has published recommendations on the security aspects of safety systems [2].

Within standards groups, there have been a number of discussion points. Safety standards groups have long argued about the appropriate Bit Error Probability value to be used for evaluating the effectiveness of a solution. Some stakeholders have suggested that the FSCP’s need to assume a higher probability of errors, with a consequent update to the algorithms used to detect these errors. A counter to this is an observation that there are many systems installed worldwide that appear to give their users an acceptable level of performance.

Furthermore, other groups have asked questions on how the performance of a system encompassing both safety and security can be understood. This paper proposes a mechanism for achieving this analysis. The fundamental needs of both safety and security systems are outlines along with the techniques used to achieve them. These are then related to the constituent parts of an automation system and a Markov model is used to describe how each of these components impacts the overall capabilities of the complete system.

Industrial Communication

Ethernet-based communication, when applied in an industrial setting, has some uniquely important needs due to the nature of industrial applications. Although these needs may vary in importance across various applications, in general they are key to supporting industrial use cases. Briefly, these are:

1. **Quick connect/disconnect of devices** – Application needs can dictate that systems are built on modules that can be connected or disconnected to account for changes in the process. An example of this is a tool changer on a robot in an automotive assembly plant. Once a tool is connected to the robot, the network devices within it need to power up and establish their connection to a controller as quickly as possible so as to facilitate a short cycle time for that part of the process.
2. **Simple integration of new devices** – A plant environment can have a wide diversity of devices. Products from different manufacturers with different functionality are often found within a given environment. As the plant evolves, new devices are brought into further productivity, output, security, or other important attributes. These devices must be integrated into the existing environment with a minimum of work and little to no downtime of the existing devices. Furthermore, the integrators of the new devices may or may not have special training, all of which contributes to the need of simple integration.
3. **Easy configuration and communication between devices** – With the wide variety of devices comes many different methods for communicating. Industrial devices support many protocols and interoperability between a group of devices is not guaranteed. Despite this, it is important that communication can be set up easily and seamlessly.
4. **Diagnostic data** – Industrial environments can sometimes be particularly challenging due to the nature of the processes (with corrosive materials for example). This contributes to equipment wearing out and requiring replacement. Usually a given environment has many different types of devices and equipment that need to communicate. For these reasons, rich diagnostic data must be available for troubleshooting, maintenance, and optimization of systems.
5. **Simple IT/OT integration** – As the OT environment continues to grow in connectivity IT systems and technology are making their way into the OT environment. With initiatives like *Industrie 4.0* gaining traction the convergence of these two environments will only continue to grow. However, OT systems have somewhat different needs, as an example downtime is generally much more serious for an OT system than an IT system. As such this trend towards integration needs to be handled carefully and with an approach that is sensitive to the unique needs of the OT environment.

CIP Safety – Requirements and Solutions

Many industrial automation applications involve high-speed moving parts, heavy loads, high temperature or pressure, corrosive or poisonous environments and frequent human machine interactions. For such applications, the protection of humans, equipment, or the environment from harm is a very important requirement. Formally defined in IEC 61508 series standard, safety is freedom from unacceptable risk and functional safety is part of the overall safety relating to the Equipment Under Control (EUC) and the EUC control system that depends on the correct functioning of the E/E/PE safety-related systems and other risk reduction measures [3].

For smaller safety systems with limited installation distance, topology, and number of devices standalone safety relays can be used with hardwired signals from sensors to actuators. The same motivations that moved communication networks into the industrial environment – greater distances, increased flexibility, reduced cost, and improved maintainability – are also driving the development of industrial safety networks [4].

For a typical industrial automation control system (IACS), it is generally unacceptable to setup one dedicated network for safety functions alongside an existing network for non-safety relevant functions due to cost, maintainability, footprint etc. Sharing one network between non-safety relevant functions and safety relevant functions in an IACS is a common practice which is adopted by most of industrial functional safety fieldbus technologies. Such practice is also referred as black channel principle, which means no safety requirement is applied to the communication system. Only the safety communication layer (SCL) on top of the communication system is responsible for the application data exchange in a safety manner.

Additional transmission of network messages with safety relevant data from one networked safety device to another networked safety device (e.g., safety sensor to safety controller) introduces extra communication related faults to the safety function. Functional safety fieldbus technologies were invented to address these extra faults and control the residual error rate of safety data to required levels.

IEC 61784-3 functional safety fieldbuses standard

Based on the black channel principle, the well-established IEC 61784-3 standard explains common principles that can be used in exchange of safety-relevant messages between participants within a distributed network in accordance with the requirements of IEC 61508 for functional safety [5]. Under the umbrella of the IEC 61784-3 generic part, Functional Safety Communication Profile 2/1 (FSCP 2/1, known as CIP Safety™) implements safety communication layer specifications on top of Communication Profile Family 2 (CPF2, known as CIP™) and Family 16 (CPF 16, known as SERCOS™). FSCP 2/1 is defined as the IEC 61784-3-2 standard.

IEC 61784-3 categorizes communication errors into corruption, unintended repetition, incorrect sequence, loss, unacceptable delay, insertion, masquerade and addressing errors, and recommends deterministic remedial measures to these communication errors, including the use of a sequence number, time stamp, time expectation, connection authentication, feedback message, data integrity assurance, redundancy with cross checking and different data integrity assurance systems [5]. The latest IEC 61784-3 Edition 4 requests that supplier of FSCP should provide proof of a sufficient overall residual error rate considering all these errors.

CIP Safety measures to detect communication errors

As an answer to address above mentioned communication errors defined in IEC 61784-3, CIP Safety™ implements corresponding measures as shown in Table 1 to detect those communication errors.

Table 1 CIP Safety™ measures to detect communication errors

Communication Errors	Measure to detect communication errors				
	Time Expectation via time stamp	Identification for sender and receiver	CRC	Redundancy with Cross Checking	Different data integrity assurance systems
Corruption	-	-	X	X	-
Unintended repetition	X	-	-	-	-
Incorrect sequence	X	-	-	-	-
Loss	X	-	-	-	-
Unacceptable delay	X	-	-	-	-
Insertion	X	X	-	-	-
Masquerade	X	X	X	X	X
Addressing	-	X	-	-	-

Data corruption detection by CRC

Of all the communication errors, data corruption is the most critical and was once the only error type required to be calculated for residual error rate, since data corruption is quite common in open networks and undetected data corruption would directly change the safety data used by safety application and cause hazards.

A Cyclic Redundancy Check (CRC) mechanism has been broadly adopted in communication protocols for its simplicity, efficiency, easy implementation, and good protection against burst type electromagnetic interference (EMI). IEC 61784-3 Annex B gives a black channel model for data integrity calculations through CRC based on a binary symmetric model, which is recommended unless a different model can be proven more applicable for a particular FSCP [5]. Based on the assumption that a Bit Error Probability (Pe) of 10^{-4} in the presence of continuous electromagnetic interference would lead to a stop of communication, after applying a safety factor of 100, IEC 61784-3 requires to use 10^{-2} as Pe for residual error probability calculation of CRC polynomials.

Though thought to be conservative, the 10^{-2} Pe requirement is still being criticized for some reasons, and higher Pe such as 0.5 was proposed to be used for CRC residual error probability calculation. The criticism is based on conceptual analysis of the communication techniques and the possibility that corruption of several bits of the message (e.g., by EMI) on the wire would result in a totally corrupted safety protocol unit (SPDU) to the safety communication layer (SCL). A SCL designed for 10^{-2} Pe would then not be able to detect such error with sufficiently high probability. Error patterns resulted from bit stuffing/destuffing, bit slipping, symbol coding/decoding, buffer overwritten are some examples.

Meanwhile, it is well understood and accepted that safety measures themselves cannot address intentional attacks. For example, without any extra protection, a safety message can be very easily altered and manipulated by an attacker but still seem to be valid to the safety data receiver. Hence cyber security measures should be deployed in open networks to ensure safety measures applicable. Cyber security works with hashing, encryption, digital signature, which are typically achieved through extensive confusion and diffusion, so the concerned behavior of bit error propagation is the design goal of some security algorithms. Along with the increasing deployment of security in communication systems, the concern of interference or contradiction between safety and security arose.

CIP Security – Requirements and Solutions

Security for industrial communication brings some additional requirements. Although the full threat analysis and mitigation of a given system is generally unique and cannot be easily summarized, some general requirements do exist. For one, devices need to have some type of authentication scheme applied. That is, it is critical that only authorized endpoints are permitted to communicate within the system. This can of course be realized through a number of different mechanisms, but the basic idea of authentication is critical to the security of the system. Once authentication has occurred, the data in transit still needs protections. Broadly, these protections are usually grouped into one of two categories, either data confidentiality or data authenticity (note some schemes can provide both through a single mechanism). This prevents an attacker from modifying data in transit and from reading data in transit. There are some situations in which data confidentiality is not strictly needed, for example if the data in transit is not considered to be high value intellectual property, or data inspection by another entity is required. Despite this, data authenticity is generally required, as preventing modification of data in transit applies in nearly every situation.

CIP Security realizes these protections through the use of the TLS and DTLS protocol, applied to EtherNet/IP. Authentication of endpoints is done via either X.509 certificates or Pre-Shared Keys (PSKs). The TLS/DTLS cipher suite chosen covers the data authenticity or data confidentiality. TLS and DTLS have allowed for a wide variety of cipher suites which provide different cryptographic algorithms for data confidentiality and data authenticity. Even within CIP Security, there are several options of cipher suites that must be supported, and all other TLS/DTLS cipher suites can optionally be supported. For the purposes of this paper the focus is on two very common algorithms: AES-CBC for data confidentiality and SHA-256 HMAC for data authenticity. These algorithms are both present in the mandatory cipher suites in CIP Security and are widely used in Internet-based TLS. Both of these algorithms provide robust information assurance properties, and are not known to have any practical, exploitable design weaknesses.

Data Protection

One area where requirements of safety and security cross over is in data protection. Security dictates requirements around data authenticity, or more specifically that it is infeasible for an attacker to modify data in transit without the modification being easily detectable. Safety dictates requirements that accidental or environmental modifications of packets are detected and do not allow the system to enter an unsafe state. These requirements are strikingly similar, the main difference being that one is focused on an ”intelligent actor” attempting to orchestrate an particular outcome and the other is more focused on “natural causes” that inhibit desired functionality. Regardless, safety and security protocols, including CIP Safety and CIP Security, both achieve these protections via very similar means. In both cases an algorithm is applied to the data payload that is used to determine the integrity of the payload. However, there are some important differences.

Safety Protection – CRC

Safety protocols like CIP Safety often apply a Cyclic Redundancy Check (CRC) to the data in transit. The CRC calculation is meant to detect when the packet has been modified. That is, when the data is created a CRC is calculated. The data is sent along with the CRC, and when the data is received the CRC calculation is again done. If the calculated CRC does not match the received CRC, then the packet is considered modified and is not acted upon.

The length of the CRC determines the probability of a “collision”, that is, an event where the data packet is different from the original yet when the CRC is calculated it still passes. A longer CRC reduces the probability of a collision, but at the expense of increased processing time and packet size.

As shown in Figure 1, the probability of a collision is called residual error probability, and the residual error probability of a specific CRC polynomial in safety domain depends on the bit error probability (Pe) and data length of the safety protocol unit (SPDU). For a high bit Pe close to 0.5, the residual error probability of a proper CRC polynomial approaches to 2^{-r}, where r is the bit length of the CRC polynomial. An improper CRC polynomial would break the upper limit of 2^{-r}.

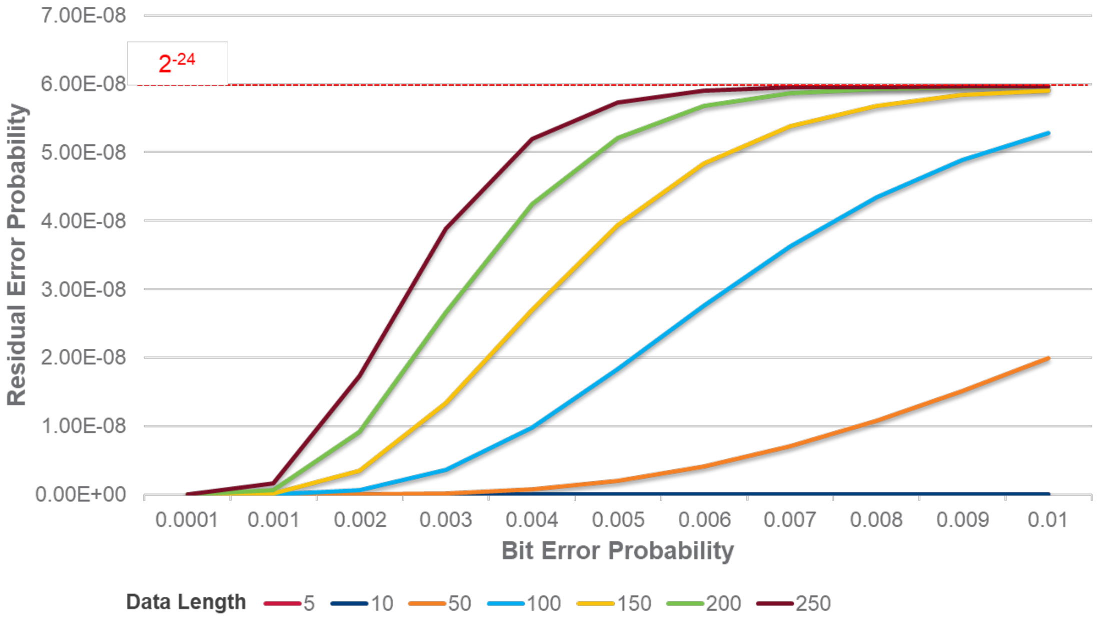

Figure 1 Example residual error probability of a 24-bit CRC polynomial

Security Protection – SHA-HMAC

The SHA-256 HMAC is similar to the CRC, at least from a high-level perspective. This is also calculated when the data is created and transported along with the data, to be again calculated by the recipient. One major difference though is that the SHA-256 HMAC relies on a secret key which only the sender and receiver know. For TLS and DTLS, this key is generally derived by both parties through the handshake, using an algorithm that an eavesdropper would not be able to reverse due to some secret knowledge (e.g. private keys) that the sender and receiver have. This is one of the main differences between the CRC and the SHA-HMAC; the lack of the secret keys for the CRC means that anyone with the original data can calculate/verify the CRC, whereas with the HMAC the secret key must be known to calculate or verify the HMAC. Beyond this, the HMACs used in security protocols, like SHA-256 HMAC, are generally of a longer length than the CRCs, so as to prevent a “brute force” attack where an attacker attempts guesses to find a collision. At the same time, the HMAC can help the security layer to detect authenticity errors before they reach the safety layer.

Collision Probability – Birthday Paradox

One way of analyzing the strength of a data integrity algorithm is through a mathematical method known as the “Birthday Paradox”. Here a brief overview of the birthday paradox is given, but for a more detailed description of all the underlying math please refer to the many publicly available resources on this idea (for example [6]). The name comes from the idea that in a given group of people, what is the probability that two of them share a birthday (note for this thought experiment it is assumed that birthdays are randomly uniformly distributed). One might assume that the probability of this occurring would be 1/365, as there are 365 unique days in a given year (ignoring leap years as a simplification). However, this naive view turns out to be false. Instead that is the probability that someone has a birthday on a chosen day, not that any two people have the same birthday, hence the term “paradox” (note this is not a true logical paradox, but rather a paradox in the sense that it defies common intuition). Math shows that it is actually much more likely to find two people with matching birthdays, in fact in a room of 23 people there is about a 50% probability that two people share the same birthday.

However, the reasoning behind this can be generalized to any type of collision, rather than looking for birthday collisions a CRC or hash collisions can be sought. For an n bit hash or CRC function, collisions become likely (50% probability) after $2^{\frac{n}{2}}$ messages. This is a good guidepost for the probability of a collision occurring in a safety system, or in a security system with an unsophisticated attacker (that is, brute force). However, for security there are other important considerations, such as whether or not there is any structure weakness in the hash function that allows an intelligent attacker a pathway to finding a collision more efficiently than simply guessing.

However, applying the birthday paradox naively to the CRCs and SHA-256 HMAC yields a simple result. For this case, the standard Ethernet CRC of 32 bits is used, of course SHA-256 HMAC has a bit length of 256.

Ethernet: $2^{\frac{32}{2}} = 2^{16} = 65536$ messages

SHA-256 HMAC: $2^{\frac{256}{2}} = 2^{128} = 3.402 \times 10^{38}$ messages

The above arguments show that the SHA-256 HMAC provides significantly more robust collision resistance properties than the CRC. In fairness the CRC bit length could be increased to 256 bits to provide similar properties, but SHA-256 is already applied through the CIP Security (TLS/DTLS) protocol. Therefore, the conclusion here is simply that the addition of the TLS/DTLS HMAC on top of the existing safety and/or Ethernet CRC simply adds to the data integrity benefit of any data transmitted, and certainly does not do anything to reduce the effectiveness of the CIP Safety protections.

Data Encryption

Encryption of data is a common technique to protect the confidentiality of that data. This is often applied via security protections, although there is no close analogue within safety systems as confidentiality of information is not a primary concern, and often not a concern at all. Encryption is generally realized through a standard, internationally recognized encryption algorithm, such as AES. Although there are many schemes in which the AES algorithm can be used, they all rely on the same basic algorithm for protection of data confidentiality. AES, and other modern encryption algorithms, have a property known as “diffusion”. Diffusion has to do with the idea that information is diffused through the encryption algorithm [7] One way of thinking of this is that when data is encrypted, each bit has an equal probability of remaining as its current value or switching to the opposite value, resulting in the cipher text appearing to be completely random (note this describes an “ideal” encryption algorithm, a real-world one will of course not have perfectly balanced probabilities). The idea of diffusion is important because if encryption is used in a safety system it will have an effect on the safety payload in transit. That is, whereas most black channel communication simply encapsulates or encodes the safety payload, encryption will significantly change the value of the payload, at least while in transit. Of course, the encryption is reversible, and will be reversed upon receipt of the packet. However, the encryption of safety data is at least a consideration for the black channel safety argument, given that so much of the safety payload is changed. However, the property of diffusion implies that if there is a bit change in the ciphertext that the plaintext, when decrypted, will be essentially random data, likely caught by the HMAC. Therefore, the diffusion property of encryption will render black channel arguments no better nor worse.

System View

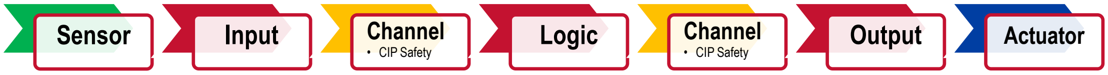

Figure 2 Safety Function

As shown in Figure 2, a safety function typically consists of a chain of safety-rated devices from a safety sensor to a logic element (safety PLC) and then to a safety actuator. Between the input device and the logic element as well as from the logic element to an output device is an industrial communication channel that can implement both security and functional safety together. Figure 3 below shows a partial implementation of that complete safety chain for the input connection side, where there is a single logical communication channel between a CIP Safety safety encoder being used as a sensing device to monitor position and velocity of a motor and a safety-rated PLC. In between them is a network switch connecting both of them through EtherNet/IP. The switch is considered part of the black channel.

The sending device implements a safety producer at a connection endpoint, while the receiving device implements a safety consumer, both using industrial communication protocols such as CIP and CIP Safety as the conduit for transmitting data from one end to the other. Figure 3 illustrates how the hardware components of that safety producer and safety consumer can be modelled from the perspective of fault detection mechanisms. These components will be used later when constructing the Markov model. The key motivation for this diagram is to show how many different encoding and detection mechanisms are used besides those provided in the safety communication layer, such as CIP Safety. This will be key to understanding the Markov Model.

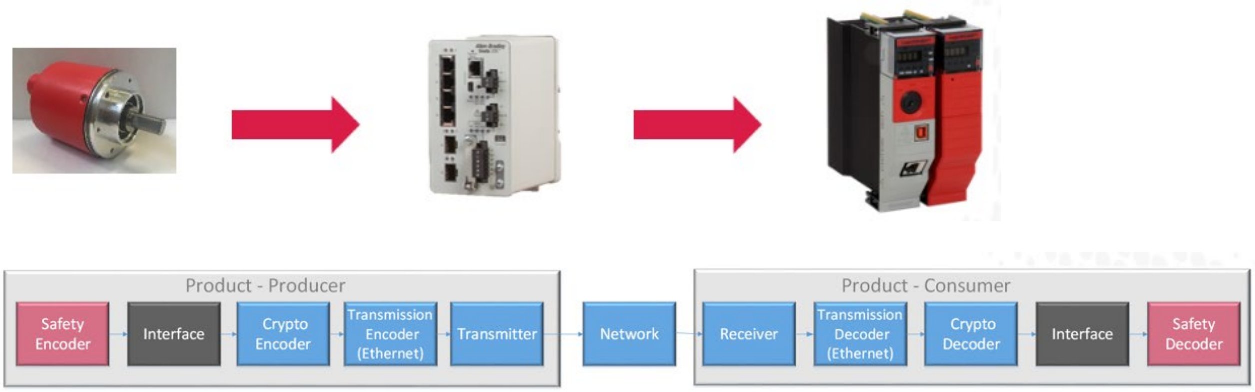

Figure 3 System View

The safety producer block is modelled by several parts. The first is the hardware and firmware that implements a Safety Encoder, not to be confused by an encoder used for monitoring position and speed of a motor. This is where safety input data is placed into a safety protocol data unit (SPDU), such as the Extended Format for CIP Safety, and thereby encoded. It is shown as a red box since it is safety related and part of the safety function. Figure 4 below shows the short and long form SPDUs of the Extended Format of CIP Safety. For the Short Form, safety encoding represents the 24-bit CRC-S5 block code and a 16-bit Time Stamp. For the Long Form, safety encoding represents the complemented safety data field, the 16-bit CRC-S3 block code over the actual data field, the 24-bit CRC-S5 block code over the complemented data field, and the Time Stamp field. Both forms also implement an implicit 16-bit Time Stamp rollover count that is included when calculating the CRC block codes. The cumulative strength of these encoding measures achieves a residual error rate that is less than 1x10^{-9} failures per hour. CIP Safety like other safety communication protocols implement end-to-end safety where spatial and temporal faults are detected and mitigated at the safety end nodes through these detection mechanisms. They are used to detect faults that occur in the Black Channel whether that be the network or the Black Channel components in the safety end devices themselves.

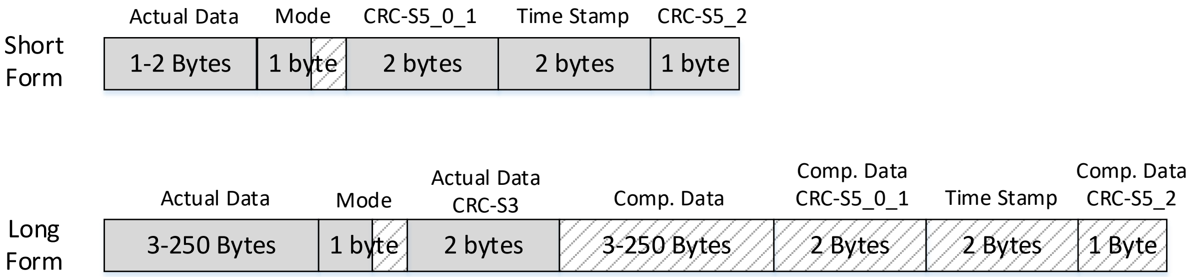

Figure 4 SPDU: CIP Safety Extended Format Encodings

The next block in the safety producer is called “Interface”. It models a communication interface or channel that is internal to the producing safety device. This block represents the transfer of the SPDU from one component or location in a device to another component for further processing by the security layer in the stack, such as CIP Security. In a real-world device this could be a serial channel or a simple memory interface for passing the SPDU from one memory location to another. This block on purpose is colored in Figure 3 as gray because from a safety context perspective it is considered part of the Black Channel even though it is technically still contained within a safety device. From this point forward the rest of the layers of the CIP communication protocol stack take that self-contained SPDU as the payload for building the rest of the CIP message around it. The SPDU is self-protected from faults introduced by these other communication layers up to a certain residual error probability.

The next block in the safety producer is the Cryptographic Encoder. This block takes the SPDU and calculates an HMAC across it or possibly encrypts the SPDU before adding the HMAC to the resultant payload. Refer to the security protection section above for more details. This is a layer of data protection above and beyond the safety protection measures but technically not a part of the safety function from the standpoint of IEC 61784-3 and IEC 61508. This measure is used to mitigate attacks from intelligent actors that purposely attempt to corrupt or masquerade message packets. But it also provides a secondary benefit against natural types of errors in the underlying hardware such as hard and soft errors in silicon.

The fourth block in the safety producer is the “Transmission Encoder”. This represents the hardware and software associated with Layer 2 encoding in the OSI communication model. For Ethernet, as an example, this entails adding a 32-bit CRC code block onto the message payload. Much like security, this provides a third layer of data protection above and beyond the safety protection measures of CIP Safety but also not considered part of the safety function in terms of spatial integrity since this is implemented within Black Channel hardware and firmware. Unlike security, though, the scope of this block code is for targeting natural sources of errors that occur on the network or the transmitter and receiver. The main purpose is to provides fault tolerance to the system against naturally occurring phenomena, in contrast to safety encoding in the SPDUs which are there to achieve a SIL rating so that no errant data is applied to a safety function.

The fifth and final block in the safety producer is the transmitter which implements Layer 1, the Physical layer, where message packets are converted to a signal that is native to the network. A transmitter can potentially introduce faults and bit flips onto an outgoing message based on some error rate specific to its semiconductor technology. The network encompasses all Black Channel devices such as routers and switches. These devices are typically not safety rated. This may include one or many intermediary hops between two safety endpoints.

The safety consumer is modelled similarly to the safety producer but in reverse both in terms of order and operation. Instead of encoding a message, it decodes a message and verifies the integrity of the message along each step moving back up the communication software stack. The first block is the receiver. This is Layer 1, the Physical Layer, in the safety consumer that converts a network native signal (e.g. electrical, wireless, light) to an equivalent digital representation of the message. Like the transmitter, hard or soft errors can be introduced onto the message through corruption of one or many bits.

After Layer 1, the message is passed to Layer 2 in the communication stack, as represented by the “Transmission Decoder” block in Figure 3 above. This is where Layer 2 detection mechanisms are checked to ensure proper data integrity. Continuing with the example of Ethernet, this block will verify the data integrity by calculating a 32-bit CRC across the payload and comparing it to the CRC value that was sent along in the message. Since this layer is Black Channel even within a safety target, this integrity measure is not safety related from a functional safety perspective but used to provide underlying reliability in message transmission.

The next block in the safety consumer is the “Crypto Decoder”. It will authenticate the message by verifying the corresponding HMAC associated with the received message and possibly decrypt the message, if need be. Like Layer 2, this is considered part of the Black Channel but nonetheless provides data integrity across the message, even if it is not formally recognized as a diagnostic measure for the safety function.

The “Interface” block in the safety consumer works in reverse compared to the safety producer. It is involved in moving a CIP Safety SPDU from one location to another within a device so that the CIP Safety subsystem can decode it and verify the Extended Format encoding, as shown in Figure 4. This interface can be either through shared memory or through a local communication peripheral such as SPI or UART. It is colored as gray in Figure 3 since it is also part of the Black Channel, as the SPDU is not altered or changed but merely moved between subsystems in the safety device for further processing.

The final block in the safety consumer is the “Safety Decoder”. This represents the CIP Safety subsystem that will take that SPDU from the received message and unpack the safety data for further processing by a safety application. The decoder will verify the data integrity and temporal integrity of the safety data by checking all the corresponding CRCs, Time Stamps, and rollover counts for the received message according to the type of encoding used, as agreed upon during connection establishment between the two safety end-point devices.

Markov Model

The reason for decomposing the producer and consumer in such a manner above is that it is helpful in constructing a mathematical model of the communication system from a fault behavior perspective. This requires an understanding of where bit errors can be introduced during the life of a SPDU in transmit with respect to these various detection mechanisms, Black Channel or not. A continuous Markov process is used in this paper to capture those fault and detection factors into one cohesive model. A Markov chain identifies a starting state where no faults are present to intermediary states that represent one or more faults in the blocks above, before transitioning to either a safe or dangerous state of the system. Dangerous in this context means that a corrupted message passed all data integrity checks (transmission decoding, security decoding, and safety decoding) and the payload is passed to the safety application without incident whereupon use of that safety application data could potentially result in undefined and unsafe behavior. This paper builds upon the Markov Model from [8], where errors during transmission, within the transmission decoding block, and within the safety encoding block were considered. This paper extends that to include security both in terms of probability of detection and a failure rate of the Cryptographic Decoder. This adds additional intermediary states and transitions to the simplified Markov diagram from [8].

Before delving into the Markov model, let's first begin with a definition of terms. As shown in Figure 5 below, all the random hardware faults and soft errors that occur during transmission from the Cryptographic Encoder through the Receiver can be grouped together into one monolithic failure rate, λ_{HTN}, for simplicity's sake. Next, the Transmission Decoder itself may fail, as represented by the failure rate term λ_{HTD}. This encompasses a wide range of failures such that the decoder cannot perform its function as intended. Using again the example of Ethernet, this represents the 32-bit CRC check across an Ethernet frame not being able to detect faults in a message. This could stem from many sources. A variable may have a stuck-at failure in a memory cell such that the function that performs the CRC check always returns a passing result. Or a soft error could cause control flow changes such that the corresponding CRC check is bypassed altogether. The term, λ_{HTD}, is meant to be include of all those sources in aggregation. Third, faults in the Cryptographic Decoder, λ_{HCD}, are similar in concept to that of a Transmission Decoder failure as far as causing the Cryptographic Decoder being unable to perform its intended operation properly either in part or in full; thereby, permitting a corrupted message to pass onto the safety communication layer, SCL, without incident. Finally, faults in the Interface Block are represented by λ_{INTF}. These faults, as will be shown later, will lead to the largest category of undetected faults since only a Safety Decoder can detect them. The safety communication layer is the only layer of defense against them. Interface errors occur either before cryptographic or transmission encoding, as shown on the left-hand side of Figure 5 below, or after their corresponding decoders as shown on the right-hand side. Black Channel detection mechanisms are mutually exclusive to these interface related errors as they will just propagate faults all the way on through to the SCL.

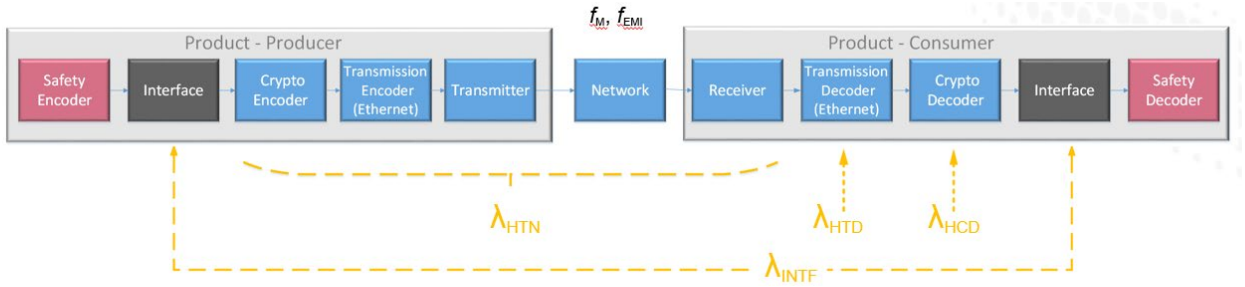

Figure 5 Failure Rate Partitioning

Table 2 Reliability Terminology

Symbol	The meaning of a Symbol
λ_{HTN}	Random HW failure rate of a transmitting device, transmission media, network components, and the receiver of a target device. This includes soft errors, such as from high energy, atmospheric particles.
λ_{HTD}	Random HW failure rate of a transmission decoder within a target device. This represents random hardware-based failures of the decoder such that it cannot perform its function, but not injection of faults into a message. This also includes soft errors resulting in bit flips, such as from high energy, atmospheric particles. **Note**: Faults in the transmission decoder that directly lead to message corruption are included mathematically within the Interface block for simplicity. In the reality, they would be

	detected by the cryptographic decoder up to a very high probability but are moved to the Interface block to keep the size of the Markov model manageable. The corresponding undetected FIT rate for this condition is not statistically significant.
λ_{HCD}	Random HW failure rate of a cryptographic decoder within a product. This represents random hardware-based failures of the decoder such that it cannot perform its function, but not the injection of faults into a message. This also includes soft errors resulting in bit flips, such as from high energy, atmospheric particles. **Note**: Faults in the cryptographic decoder that directly lead to message corruption are included mathematically within the Interface block for simplicity to keep the size of the Markov model manageable.
λ_{INTF}	Random HW failure rate within the interface between the cryptographic decoder and the safety decoder. This could represent a serial channel or a simple memory interface. This also includes soft errors resulting in bit flips, such as from high energy atmospheric particles.
f_{EMI}	Mean frequency of corrupted messages caused by EMI.
f_M	Mean frequency of messages generated by a transmitter.
p_{UT}	Probability of the transmission decoder not detecting an error.
p_{UC}	Probability of the cryptographic decoder not detecting an error.
p_{US}	Probability of the safety decoder not detecting an error.
δ_T	The intensity of the transition to a safety state caused by the detection mechanism of the transmission decoder.
δ_C	The intensity of the transition to a safety state caused by the detection mechanism of the cryptographic decoder.
δ_S	The intensity of the transition to a safety state caused by the detection mechanism of the safety decoder.

Continuing with the definition of terms, the strength of detection for each of the three decoders is also an integral part of the Markov model. This is shown in Figure 7 as δ_T, δ_C, and δ_S for transmission decoder detection, cryptographic decoder detection, and safety decoder detection, respectively. They represent the probability that each detection mechanism will detect bit errors in the received message as represented by a residual error probability p_U. The strength or intensity is typically attributable to the size or type of polynomial/algorithm used in the block code, e.g. CRC or hash. For example, based on the residual error probability of a CRC polynomial, faults in a message could be of a certain size or arrangement such that the CRC check is unable to detect message corruption. This can be mathematically represented as a worst-case residual error probability of 2^{-r} for proper CRC polynomials, where r is the number of bits of the CRC block code. Graphically, the residual error probability for each decoder is shown in the diagram as p_{UT}, p_{UC}, and p_{US}, respectively. The λ error rates include both random hardware faults as well as soft errors on volatile memory cells or gates. But they are not the only source of faults. Bit corruption in a message could also occur naturally through electromagnetic interference and crosstalk, as modelled by f_{EMI}. Finally, the residual error probability of a communication channel is not just dependent on these aforementioned factors but also is directly related to the mean frequency of messages being transmitted. Simply stated, the more often messages are received, the more likely that data corruption can occur physically either on the wire or in the end products themselves. This is shown in the Markov Model by the term f_M.

Figure 6 and Figure 7 illustrate an uncondensed version of the Markov Model. It contains 18 different states altogether, which are based on all the all the different permutations of errors to four main hardware blocks/groupings described above, $2^4 = 16$ states, plus two extra absorbing states for the Hazard and the Safe State.

- λ_{HTN}
- λ_{HTD}
- λ_{HCD}
- λ_{INTF}

2^4 = 16 states (shown graphically as States 1-16)

The columns in Figure 6 are arranged as such on purpose based on how many of the four blocks have been compromised with faults. The first column, shown only by State 1, represents the starting state where no faults are present, where all messages are pristine, and all decoders are functioning as intended. The second column, States 2-5, represent the set of states where only one of the four blocks has at least one error. The third column, States 6-11, represent the set of states where two of the four blocks have been compromised and no longer function as intended; there are six different combinations of those. The fourth column represents the set of states where three of the four blocks have been compromised and no longer work as intended either in part or full; there are four different combinations of those. The fifth column, shown only by State 16, represents all four blocks malfunctioning to some degree. These faults could occur simultaneously or could have accumulated over time if a fault in a block had not been detected through diagnostic measures running in the background of the target device.

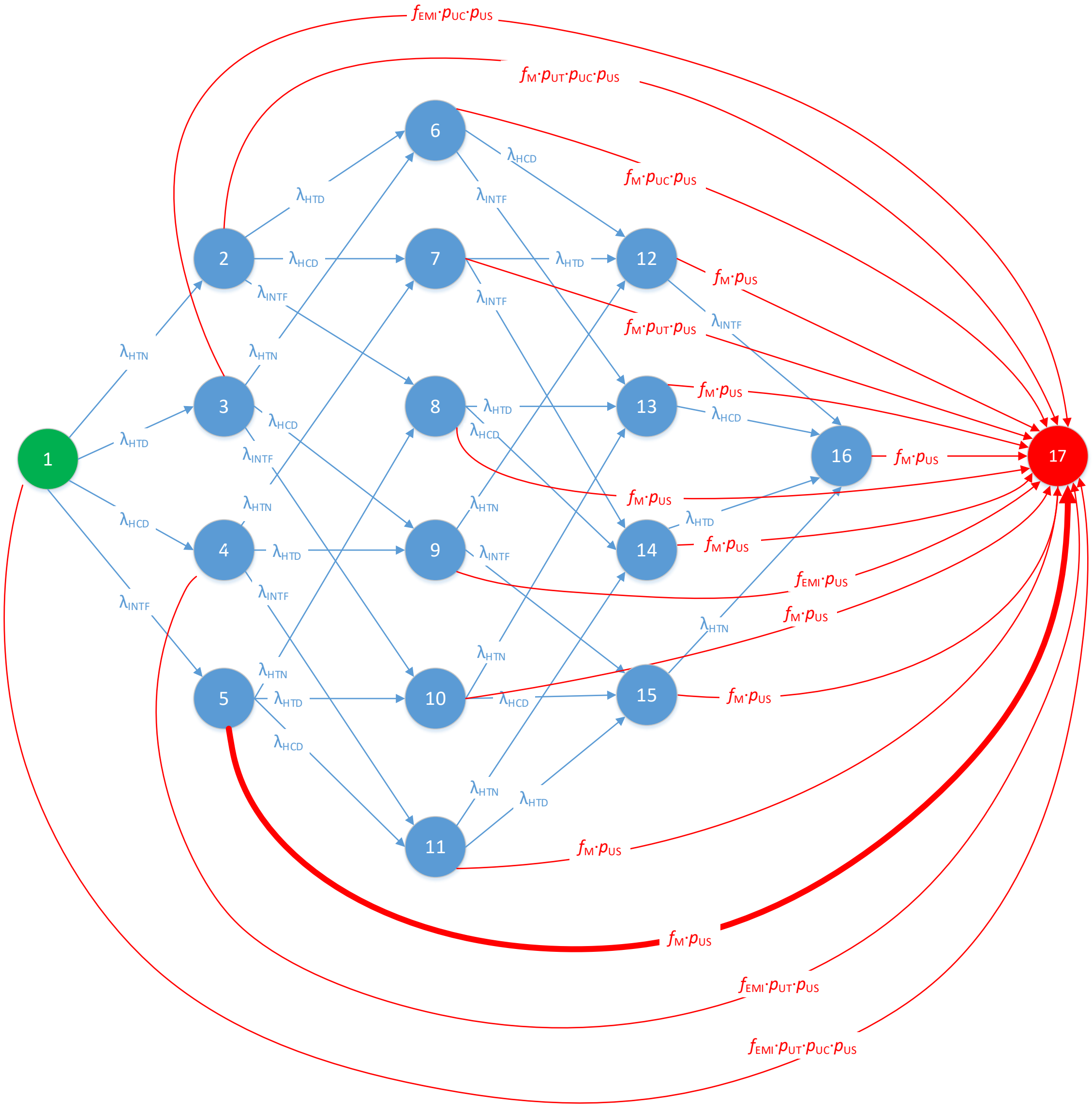

Figure 6 Uncondensed Markov Model, part 1

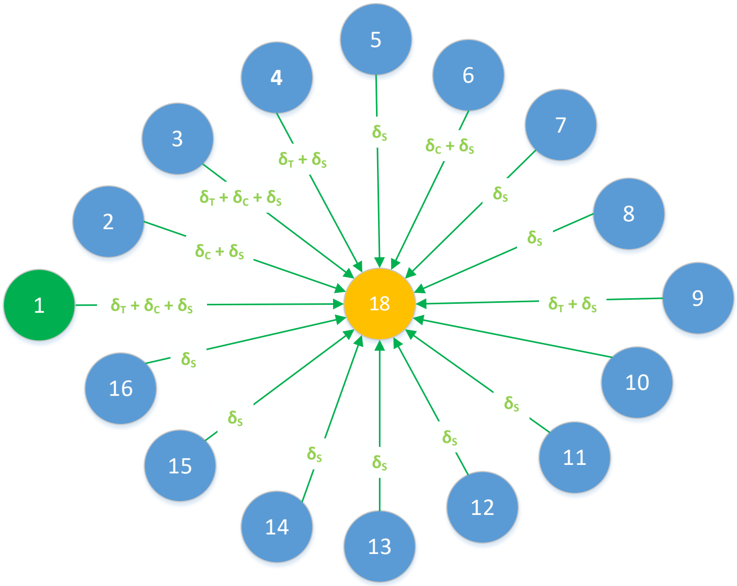

Figure 7 Uncondensed Markov Model, part 2

Since Interface block errors are independent and not detectable by the two Black Channel decoders (Transmission Decoder and Cryptographic Decoder), the uncondensed Markov Model can be collapsed and simplified down to 11 states, as shown below in Figure 8 and Figure 9. This has the effect of simplifying the total number of differential equations that are used to characterize the Markov Model – one differential equation for each state. Table 3 defines each of those 11 states and the starting probability of those states at time t=0. Like before, the states in this diagram are also arranged in columns with the exception of State 9, which represents a fault in one or two of the Interface blocks which when that happens there are only two transition arrows out of that state, either into the Safe State (State 11) or the Hazard State (State 10). This is the reason the Markov Model could be collapsed. Like before, columns 2-4 show the combinations of 1 fault, 2 faults, and 3 faults based on λ_{HTN}, λ_{HTD}, and λ_{HCD}.

As previously mentioned, each transition error in the Markov Model from one state to another represents a failure in a block based on the error rate of the corresponding λ. By examining the transitions arrow both exiting and entering a state one can calculate the failure rate, λ, at each respective state.

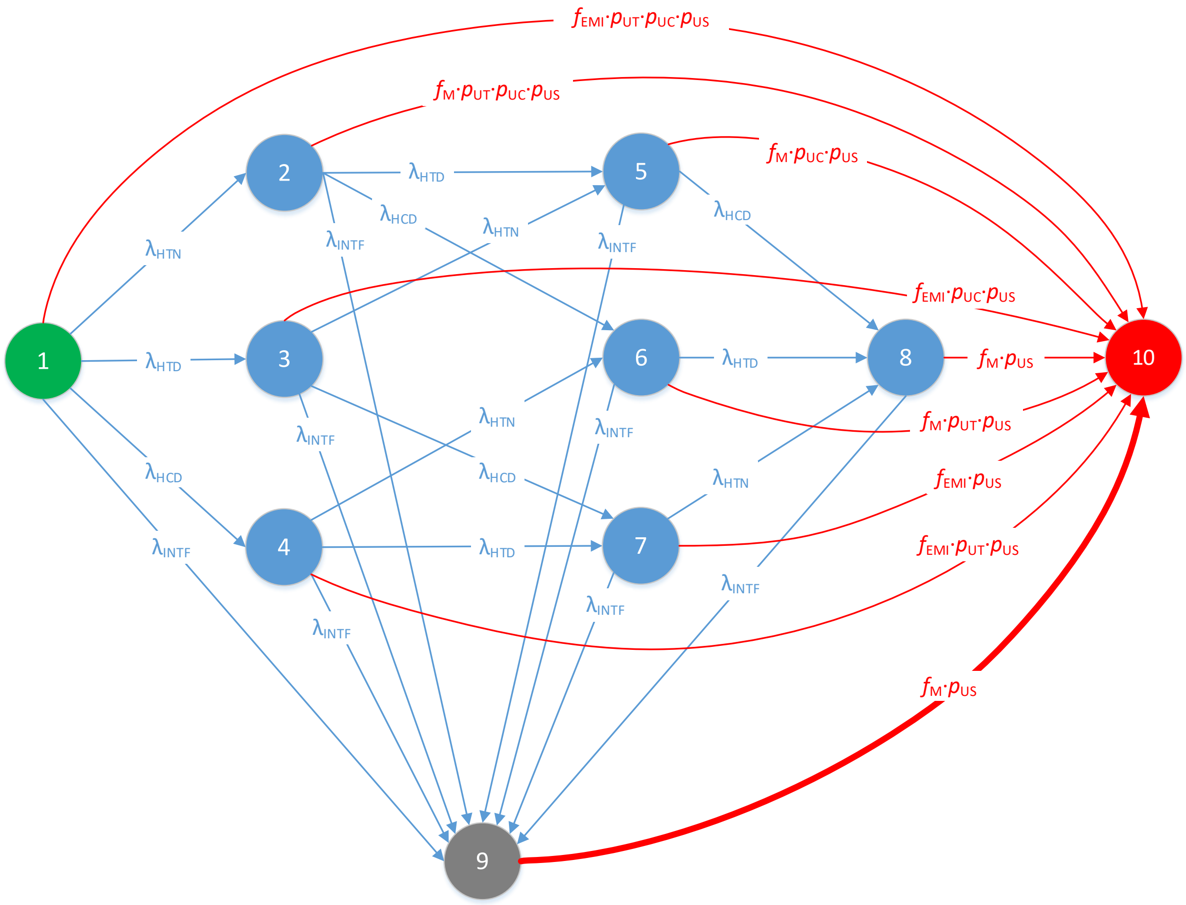

Figure 8 Condensed Markov Model, part 1

Table 3 Markov Model State Definitions

State	State Description	P(t=0)
1	The transmission system is operational without faults. Transmissions occur under the potential of EMI disturbances or soft errors.	1
2	State of the transmission system when random HW fault(s) or soft errors have occurred within either the transmitting device, the transmission media, network components, the receiver in the target device, or any combination thereof.	0
3	State of the transmission system when random HW fault(s) or soft errors have occurred ONLY within the transmission decoder of a target device, such that it cannot perform its normal function, but itself does not inject faults into the message. An example of this state would be when the transmission decoder is fully bypassed.	0
4	State of the transmission system when random HW fault(s) or soft errors have occurred ONLY within the cryptographic decoder of a target device, such that it cannot perform its normal function, but itself does not inject faults into the message. An example of this state would be when the cryptographic decoder is fully bypassed.	0
5	State of the transmission system when random HW fault(s) or soft errors have occurred within either the transmitting device, the transmission media, network components, the receiver in the target	0

State	State Description	P(t=0)
	device, or any combination thereof, AND random HW fault(s) or soft errors have occurred within the transmission decoder such that it cannot perform its normal function.	
6	State of the transmission system when random HW fault(s) or soft errors have occurred within either the transmitting device, the transmission media, network components, the receiver in the target device, or any combination thereof, AND random HW fault(s) or soft errors have occurred within the cryptographic decoder such that it cannot perform its normal function.	0
7	State of the transmission system when random HW fault(s) or soft errors have occurred within both the transmission decoder AND the cryptographic decoder such that neither of them cannot perform their normal function. In this state, EMI is the only mechanism modelled to cause a corrupted message.	0
8	State of the transmission system when random HW fault(s) or soft errors have occurred within both the 1) transmission decoder AND 2) the cryptographic decoder such that neither of them cannot perform their normal function, AND 3) within either the transmitting device, transmission media, network components, the receiver of the target device, or any combination thereof. This state represents the condition when only the safety decoder is the only functional decoder, and an error has corrupted the message somewhere during transmission.	0
9	State of the transmission system when random HW fault(s) or soft errors have occurred within the interface between the cryptographic decoder and the safety decoder in either the producing or consuming device. In this state, messages will be corrupted when forwarded to the safety decoder, regardless of whether a fault has also occurred previously or not in the transmission decoder, cryptographic decoder, network, or the transmitting device.	0
10	State of the transmission system when a safety hazard is present and not detectable.	0
11	State of the transmission system when a fault has been detected and controlled by either the transmission decoder, cryptographic decoder, or safety decoder such that a safety state is entered.	0

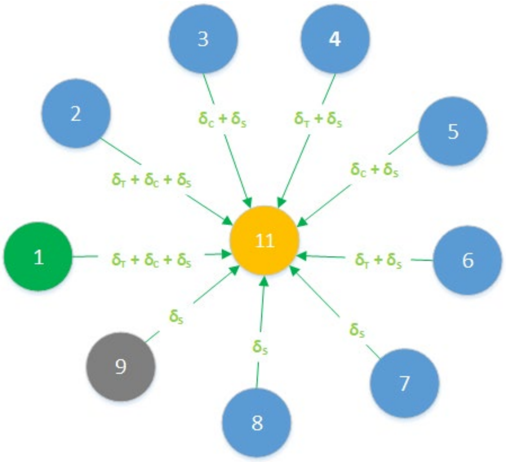

Figure 9 Condensed Markov Model, part 2

From [8] it has been shown that a Markov model is mathematically described by a set of differential equations based on the matrix multiplication as shown below:

$$\frac{dP(t)}{dt} = P(t) \cdot A \tag{1}$$

where:
$P(t) = \{p_1(t),\ p_2(t),\ p_3(t),\ p_4(t),\ p_5(t),\ p_6(t),\ p_7(t),\ p_8(t),\ p_9(t),\ p_{10}(t),\ p_{11}(t)\}$

is a vector of probabilities of being in each state of the Markov model with initial probabilities of P(t=0) = {1,0,0,0,0,0,0,0,0,0,0}

A is a transition probability matrix from one state to another, described as follows according to the transitions in and out of each state, as described by the Markov Model diagram from Figure 8 and Figure 9:

$$\text{A} = \begin{bmatrix} \lambda_1 & \lambda_{HTN} & \lambda_{HTD} & \lambda_{HCD} & 0 & 0 & 0 & 0 & \lambda_{INTF} & f_{EMI} \cdot p_{UT} \cdot p_{UC} \cdot p_{US} & \delta_T +\ \delta_C + \delta_S \\ 0 & \lambda_2 & 0 & 0 & \lambda_{HTD} & \lambda_{HCD} & 0 & 0 & \lambda_{INTF} & f_M \cdot p_{UT} \cdot p_{UC} \cdot p_{US} & \delta_T +\ \delta_C + \delta_S \\ 0 & 0 & \lambda_3 & 0 & \lambda_{HTN} & 0 & \lambda_{HCD} & 0 & \lambda_{INTF} & f_{EMI} \cdot p_{UC} \cdot p_{US} & \delta_C + \delta_S \\ 0 & 0 & 0 & \lambda_4 & 0 & \lambda_{HTN} & \lambda_{HTD} & 0 & \lambda_{INTF} & f_{EMI} \cdot p_{UT} \cdot p_{US} & \delta_T + \delta_S \\ 0 & 0 & 0 & 0 & \lambda_5 & 0 & 0 & \lambda_{HCD} & \lambda_{INTF} & f_M \cdot p_{UC} \cdot p_{US} & \delta_C + \delta_S \\ 0 & 0 & 0 & 0 & 0 & \lambda_6 & 0 & \lambda_{HTD} & \lambda_{INTF} & f_M \cdot p_{UT} \cdot p_{US} & \delta_T + \delta_S \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_7 & \lambda_{HTN} & \lambda_{INTF} & f_{EMI} \cdot p_{US} & \delta_S \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_8 & \lambda_{INTF} & f_M \cdot p_{US} & \delta_S \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_9 & f_M \cdot p_{US} & \delta_S \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

$$\frac{dP(t)}{dt} = [p_1(t) \quad p_2(t) \quad p_3(t) \quad p_4(t) \quad p_5(t) \quad p_6(t) \quad p_7(t) \quad p_8(t) \quad p_9(t) \quad p_{10}(t) \quad p_{11}(t)] \cdot$$

$$\begin{bmatrix} \lambda_1 & \lambda_{HTN} & \lambda_{HTD} & \lambda_{HCD} & 0 & 0 & 0 & 0 & \lambda_{INTF} & f_{EMI} \cdot p_{UT} \cdot p_{UC} \cdot p_{US} & \delta_T +\ \delta_C + \delta_S \\ 0 & \lambda_2 & 0 & 0 & \lambda_{HTD} & \lambda_{HCD} & 0 & 0 & \lambda_{INTF} & f_M \cdot p_{UT} \cdot p_{UC} \cdot p_{US} & \delta_T +\ \delta_C + \delta_S \\ 0 & 0 & \lambda_3 & 0 & \lambda_{HTN} & 0 & \lambda_{HCD} & 0 & \lambda_{INTF} & f_{EMI} \cdot p_{UC} \cdot p_{US} & \delta_C + \delta_S \\ 0 & 0 & 0 & \lambda_4 & 0 & \lambda_{HTN} & \lambda_{HTD} & 0 & \lambda_{INTF} & f_{EMI} \cdot p_{UT} \cdot p_{US} & \delta_T + \delta_S \\ 0 & 0 & 0 & 0 & \lambda_5 & 0 & 0 & \lambda_{HCD} & \lambda_{INTF} & f_M \cdot p_{UC} \cdot p_{US} & \delta_C + \delta_S \\ 0 & 0 & 0 & 0 & 0 & \lambda_6 & 0 & \lambda_{HTD} & \lambda_{INTF} & f_M \cdot p_{UT} \cdot p_{US} & \delta_T + \delta_S \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_7 & \lambda_{HTN} & \lambda_{INTF} & f_{EMI} \cdot p_{US} & \delta_S \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_8 & \lambda_{INTF} & f_M \cdot p_{US} & \delta_S \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_9 & f_M \cdot p_{US} & \delta_S \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

The left-hand side can be specified as follows:

$$\frac{dP(t)}{dt} = \begin{bmatrix} \frac{dp_1(t)}{dt} & \frac{dp_2(t)}{dt} & \frac{dp_3(t)}{dt} & \frac{dp_4(t)}{dt} & \frac{dp_5(t)}{dt} & \frac{dp_6(t)}{dt} & \frac{dp_7(t)}{dt} & \frac{dp_8(t)}{dt} & \frac{dp_9(t)}{dt} & \frac{dp_{10}(t)}{dt} & \frac{dp_{11}(t)}{dt} \end{bmatrix} \tag{4}$$

State 1 represents no present errors in the four major grouping blocks (λ_{HTN}, λ_{HTD}, λ_{HCD}, λ_{INTF}) of the communication system but a message in transit could still be subject to electromagnetic interference at a frequency of occurrence of f_{EMI}. The dangerous failure rate from State 1 to State 10 is equal to all three decoders concurrently not detecting a fault based on their respective residual error probabilities (p_{UT} ·

$p_{UC} \cdot p_{US}$) times the frequency, f_{EMI}, of bit corruption to a message per hour based on electromagnetic interference and signal crosstalk. f_{M} is not considered in this transition to State 10 because there are no known faults in the transmission block hardware from the Cryptographic Encoder through the network to the receiver. Nor are there any faults in the Interface blocks, either. In this state the source of fault introduction is EMI only. The matrix multiplication yields the following differential equation for State 1.

$$\lambda_1 = -\left[\lambda_{HTN} + \lambda_{HTD} + \lambda_{HCD} + \lambda_{INTF} + f_{EMI} \cdot p_{UT} \cdot p_{UC} \cdot p_{US} + \delta_T + \delta_C + \delta_S\right] \quad (5)$$

$$\frac{dp_1(t)}{dt} = \lambda_1 \cdot p_1(t) \quad (6)$$

State 2 models a condition where the only the block (λ_{HTN}) that has experienced an error is the one representing the transmitting device, transmission media, network components. Fortunately, in this state all three decoders are still intact and operational and could potentially detect a fault in an SPDU up to some probability based on the strength of each respective block code. Therefore, the dangerous failure rate from State 2 to State 10 is equal to all three decoders concurrently not detecting a fault based on their respective residual error probabilities ($p_{UT} \cdot p_{UC} \cdot p_{US}$) times the mean frequency, f_{M}, of messages. EMI related errors are not considered statistically significant in comparison to already existing errors, λ_{HTN}, from the transmission block that will manifest itself far more often based on a rate of f_{M} messages per hour. Therefore, EMI errors are not included in the equation below for simplicity's sake. The matrix multiplication yields the following differential equation for State 2.

$$\lambda_2 = -\left[\lambda_{HTD} + \lambda_{HCD} + \lambda_{INTF} + f_M \cdot p_{UT} \cdot p_{UC} \cdot p_{US} + \delta_T + \delta_C + \delta_S\right] \quad (7)$$

$$\frac{dp_2(t)}{dt} = \lambda_{HTN} \cdot p_1(t) + \lambda_2 \cdot p_2(t) \quad (8)$$

State 3 models a condition where the only the block of the four that has experienced an error is the Transmission Decoder such that it cannot perform its function as intended. In this state only the Cryptographic Decoder and Safety Decoder are still intact and operational and could potentially detect a fault in an SPDU up to some probability based on the strength of each respective block code. Therefore, the dangerous failure rate from State 3 to State 10 is equal to both of those decoders concurrently not detecting a fault based on their respective residual error probabilities ($p_{UC} \cdot p_{US}$) times the frequency, f_{EMI}, of bit corruption to a message per hour based on electromagnetic interference and signal crosstalk. f_{M} is not considered in this transition to State 10 because there are no known faults in the transmission block hardware from the Cryptographic Encoder through the network to the receiver. Nor are there any faults in the Interface blocks, either. In this state the source of fault introduction is EMI only. Note: if and when transmission block faults do occur later, then that would result in a transition within the Markov model from State 3 to State 5 and thereby a different differential equation is applicable where f_{M} then becomes statistically dominant at that point. The matrix multiplication yields the following differential equation for State 3.

$$\lambda_3 = -\left[\lambda_{HTN} + \lambda_{HCD} + \lambda_{INTF} + f_{EMI} \cdot p_{UC} \cdot p_{US} + \delta_C + \delta_S\right] \quad (9)$$

$$\frac{dp_3(t)}{dt} = \lambda_{HTD} \cdot p_1(t) + \lambda_3 \cdot p_3(t) \quad (10)$$

State 4 models a condition where the only the block of the four that has experienced an error is the Cryptographic Decoder such that it cannot perform its function as intended. In this state only the Transmission Decoder and Safety Decoder are still intact and operational and could potentially detect a fault in an SPDU up to some probability based on the strength of each respective block code. Therefore, the dangerous failure rate from State 4 to State 10 is equal to both of those decoders concurrently not detecting a fault based on their respective residual error probabilities ($p_{UT} \cdot p_{US}$) times the frequency, f_{EMI}, of bit corruption to a message per hour based on electromagnetic interference and signal crosstalk. f_{M} is not considered in this transition to State 10 because there are no known faults in the transmission block hardware from the Cryptographic Encoder through the network to the receiver. Nor are there any faults in the Interface blocks, either. In this state the source of fault introduction is EMI only. Note: if and when transmission block faults do occur later, then that would result in a transition within the Markov model from State 4 to State 6 and thereby a different differential equation is applicable where f_M then becomes statistically dominant at that point. The matrix multiplication yields the following differential equation for State 4.

$$\lambda_4 = -\left[\lambda_{HTN} + \lambda_{HTD} + \lambda_{INTF} + f_{EMI} \cdot p_{UT} \cdot p_{US} + \delta_T + \delta_S\right] \tag{11}$$

$$\frac{dp_4(t)}{dt} = \lambda_{HCD} \cdot p_1(t) + \lambda_4 \cdot p_4(t) \tag{12}$$

State 5 models a condition where two of the four blocks have experienced an error: a) the Transmission Decoder such that it cannot perform its function as intended, and b) the block representing the transmitting device, transmission media, and network components such that a message can be corrupted in transit. In this state only the Cryptographic Decoder and Safety Decoder are still intact and operational and could potentially detect a fault in an SPDU up to some probability based on the strength of each respective block code. Therefore, the dangerous failure rate from State 5 to State 10 is equal to both of those decoders concurrently not detecting a fault based on their respective residual error probabilities ($p_{UC} \cdot p_{US}$) times the mean frequency, f_M, of messages. EMI related errors are not considered statistically significant in comparison to already existing errors, λ_{HTN}, from the transmission block that will manifest itself far more often based on a rate of f_M messages per hour. Therefore, EMI errors are not included in the equation below for simplicity's sake. The matrix multiplication yields the following differential equation for State 5.

$$\lambda_5 = -\left[\lambda_{HCD} + \lambda_{INTF} + f_M \cdot p_{UC} \cdot p_{US} + \delta_C + \delta_S\right] \tag{13}$$

$$\frac{dp_5(t)}{dt} = \lambda_{HTD} \cdot p_2(t) + \lambda_{HTN} \cdot p_3(t) + \lambda_5 \cdot p_5(t) \tag{14}$$

State 6 models a condition where two of the four blocks have experienced an error: a) the Cryptographic Decoder such that it cannot perform its function as intended, and b) the block representing the transmitting device, transmission media, and network components such that a message can be corrupted in transit. In this state only the Transmission Decoder and Safety Decoder are still intact and operational and could potentially detect a fault in an SPDU up to some probability based on the strength of each respective block code. Therefore, the dangerous failure rate from State 6 to State 10 is equal to both of those decoders concurrently not detecting a fault based on their respective residual error probabilities ($p_{UT} \cdot p_{US}$) times the mean frequency, f_M, of messages. EMI related errors are not considered statistically significant in comparison to already existing errors, λ_{HTN}, from the transmission block that will manifest itself far more often based on a rate of f_M messages per hour. Therefore, EMI errors are not included in the equation below for simplicity's sake. The matrix multiplication yields the following differential equation for State 6.

$$\lambda_6 = -\left[\lambda_{HTD} + \lambda_{INTF} + f_M \cdot p_{UT} \cdot p_{US} + \delta_T + \delta_S\right] \tag{15}$$

$$\frac{dp_6(t)}{dt} = \lambda_{HCD} \cdot p_2(t) + \lambda_{HTN} \cdot p_4(t) + \lambda_6 \cdot p_6(t) \tag{16}$$

State 7 models a condition where two of the four blocks have experienced an error: a) the Cryptographic Decoder and b) Transmission Decoder such that both decoders cannot perform their respective function as intended. In this state only the Safety Decoder is still intact and operational and could potentially detect a fault in an SPDU up to some probability based on the strength of its respective block code and other safety encoding measures. Therefore, the dangerous failure rate from State 7 to State 10 is equal the safety decoder not detecting a fault based on its residual error probability, p_{US}, times the frequency, f_{EMI}, of bit corruption to a message per hour based on electromagnetic interference and signal crosstalk. f_M is not considered in this transition to State 10 because there are no known faults in the transmission block hardware from the Cryptographic Encoder through the network to the receiver. In this state the source of fault introduction is EMI only. Note: if and when transmission block faults do occur later, then that would result in a transition within the Markov model from State 7 to State 8 and thereby a different differential equation is applicable where f_M then becomes statistically dominant at that point. The matrix multiplication yields the following differential equation for State 7.

$$\lambda_7 = -[\lambda_{HTN} + \lambda_{INTF} + f_{EMI} \cdot p_{US} + \delta_S] \tag{17}$$

$$\frac{dp_7(t)}{dt} = \lambda_{HCD} \cdot p_3(t) + \lambda_{HTD} \cdot p_4(t) + \lambda_7 \cdot p_7(t) \tag{18}$$

State 8 models a condition where all four blocks have experienced an error. In this state only the Safety Decoder is still intact and operational and could potentially detect a fault in an SPDU up to some probability based on the strength of its respective block code and other safety encoding measures. Therefore, the dangerous failure rate from State 8 to State 10 is equal the safety decoder not detecting a fault based on its residual error probability, p_{US}, times the mean frequency, f_M, of messages. EMI related errors are not considered statistically significant in comparison to already existing errors, λ_{HTN}, from the transmission block that will manifest itself far more often based on a rate of f_M messages per hour. Therefore, EMI errors are not included in the equation below for simplicity's sake. The matrix multiplication yields the following differential equation for State 8.

$$\lambda_8 = -[\lambda_{INTF} + f_M \cdot p_{US} + \delta_S] \tag{19}$$

$$\frac{dp_8(t)}{dt} = \lambda_{HCD} \cdot p_5(t) + \lambda_{HTD} \cdot p_6(t) + \lambda_{HTN} \cdot p_7(t) + \lambda_8 \cdot p_8(t) \tag{20}$$

State 9 models a condition where one or two of the Interface blocks has experienced an error such that messages will be corrupted when forwarded to the safety decoder. In this state both the Cryptographic Decoder and Transmission Decoder have no ability to detect faults based on the principle of garbage in and garbage out. Only the Safety Decoder has the potential to detect a fault in an SPDU up to some probability based on the strength of its respective block code and other safety encoding measures. Therefore, the dangerous failure rate from State 9 to State 10 is equal the safety decoder not detecting a fault based on its residual error probability, p_{US}, times the mean frequency, f_M, of messages. EMI related errors are not considered statistically significant in comparison to already existing errors, λ_{INTF}, from the Interface block that will manifest itself far more often based on a rate of f_M messages per hour. Therefore, EMI errors are not included in the equation below for simplicity's sake. The matrix multiplication yields the following differential equation for State 9. The matrix multiplication yields the following differential equation for State 9.

$$\lambda_9 = -[f_M \cdot p_{US} + \delta_S] \tag{21}$$

$$\begin{aligned}\frac{dp_9(t)}{dt} = {} & \lambda_{INTF} \cdot p_1(t) + \lambda_{INTF} \cdot p_2(t) + \lambda_{INTF} \cdot p_3(t) + \lambda_{INTF} \cdot p_4(t) + \\ & \lambda_{INTF} \cdot p_5(t) + \lambda_{INTF} \cdot p_6(t) + \lambda_{INTF} \cdot p_7(t) + \lambda_{INTF} \cdot p_8(t) + \lambda_9 \cdot p_9(t)\end{aligned} \tag{22}$$

State 10 models the Hazard State where a corrupted message passed all data integrity checks (transmission decoding, security decoding, and safety decoding) and the payload is passed to the safety application without incident whereupon use of that safety application data could potentially result in undefined and unsafe behavior. It is an absorbing state with no transitions out of it, mathematically. The matrix multiplication yields the following differential equation for State 10.

$$\frac{dp_{10}(t)}{dt} = f_{EMI} \cdot p_{UT} \cdot p_{UC} \cdot p_{US} \cdot p_1(t) + f_M \cdot p_{UT} \cdot p_{UC} \cdot p_{US} \cdot p_2(t) + \tag{23}$$

$$f_{EMI} \cdot p_{UC} \cdot p_{US} \cdot p_3(t) + f_{EMI} \cdot p_{UT} \cdot p_{US} \cdot p_4(t) + f_M \cdot p_{UC} \cdot p_{US} \cdot p_5(t) + \\ f_M \cdot p_{UT} \cdot p_{US} \cdot p_6(t) + f_{EMI} \cdot p_{US} \cdot p_7(t) + \ f_M \cdot p_{US} \cdot p_8(t) + f_M \cdot p_{US} \cdot p_9(t)$$

State 11 models the Safe State where one of the three data integrity checks (transmission decoding, security decoding, and safety decoding) had detected a corrupted message and the communication object state machine had transitioned to a Safe State and had closed the safety connection and rejected the received message. The matrix multiplication yields the following differential equation for State 11.

$$\begin{aligned} \frac{dp_{11}(t)}{dt} = {} & (\delta_T + \ \delta_C + \delta_S) \cdot p_1(t) + \ (\delta_T + \ \delta_C + \delta_S) \cdot p_2(t) + (\delta_C + \delta_S) \cdot p_3(t) + \\ & (\delta_T + \delta_S) \cdot p_4(t) + (\delta_C + \delta_S) \cdot p_5(t) + (\delta_T + \delta_S) \cdot p_6(t) + \delta_S \cdot p_7(t) + \\ & \delta_S \cdot p_8(t) + \delta_S \cdot p_9(t) \end{aligned} \tag{24}$$

Solving these 11 differential equations will result in the probability matrix for each state, $p_1(t)$ through $p_{11}(t)$. The final residual error of the hazard state, State 10, contains many terms, most of which have many multiplicative factors which significantly reduce the bit error probability as seen by the SCL far below Pe=10^{-2}. The term which becomes most dominant is one in which is not influenced by the security system, but rather is due to the Interface blocks between the safety and security subsystems. Furthermore, the λ_{INTF} failure rate for the Interface blocks can be controlled against systematic anomalies since the end products are not only compliant to the systematic capability levels of IEC 61508, but also similar capabilities included by IEC 62443 across the entire product and not just the safety subsystem. This would include handling of systematic type of faults like buffer overflows or improper range checking that could occur across the Interface block.

$$\begin{aligned}
\frac{dp_1(t)}{dt} &= \lambda_1 \cdot p_1(t) \\
\frac{dp_2(t)}{dt} &= \lambda_{HTN} \cdot p_1(t) + \ \lambda_2 \cdot p_2(t) \\
\frac{dp_3(t)}{dt} &= \lambda_{HTD} \cdot p_1(t) + \ \lambda_3 \cdot p_3(t) \\
\frac{dp_4(t)}{dt} &= \lambda_{HCD} \cdot p_1(t) + \ \lambda_4 \cdot p_4(t) \\
\frac{dp_5(t)}{dt} &= \lambda_{HTD} \cdot p_2(t) + \ \lambda_{HTN} \cdot p_3(t) + \ \lambda_5 \cdot p_5(t) \\
\frac{dp_6(t)}{dt} &= \lambda_{HCD} \cdot p_2(t) + \ \lambda_{HTN} \cdot p_4(t) + \ \lambda_6 \cdot p_6(t) \\
\frac{dp_7(t)}{dt} &= \lambda_{HCD} \cdot p_3(t) + \ \lambda_{HTD} \cdot p_4(t) + \ \lambda_7 \cdot p_7(t) \\
\frac{dp_8(t)}{dt} &= \lambda_{HCD} \cdot p_5(t) + \ \lambda_{HTD} \cdot p_6(t) + \lambda_{HTN} \cdot p_7(t) + \lambda_8 \cdot p_8(t) \\
\frac{dp_9(t)}{dt} &= \lambda_{INTF} \cdot p_1(t) + \ \lambda_{INTF} \cdot p_2(t) + \lambda_{INTF} \cdot p_3(t) + \lambda_{INTF} \cdot p_4(t) + \lambda_{INTF} \cdot p_5(t) + \lambda_{INTF} \cdot p_6(t) \\
&\quad + \lambda_{INTF} \cdot p_7(t) + \lambda_{INTF} \cdot p_8(t) + \lambda_9 \cdot p_9(t) \\
\frac{dp_{10}(t)}{dt} &= f_{EMI} \cdot p_{UT} \cdot p_{UC} \cdot p_{US} \cdot p_1(t) + \ f_M \cdot p_{UT} \cdot p_{UC} \cdot p_{US} \cdot p_2(t) + f_{EMI} \cdot p_{UC} \cdot p_{US} \cdot p_3(t) + f_{EMI} \\
&\quad \cdot p_{UT} \cdot p_{US} \cdot p_4(t) + f_M \cdot p_{UC} \cdot p_{US} \cdot p_5(t) + f_M \cdot p_{UT} \cdot p_{US} \cdot p_6(t) + f_{EMI} \cdot p_{US} \cdot p_7(t) \\
&\quad + f_M \cdot p_{US} \cdot p_8(t) + f_M \cdot p_{US} \cdot p_9(t)
\end{aligned} \tag{25}$$

$$\frac{dp_{11}(t)}{dt} = (\delta_T + \delta_C + \delta_S) \cdot p_1(t) + (\delta_T + \delta_C + \delta_S) \cdot p_2(t) + (\delta_C + \delta_S) \cdot p_3(t) + (\delta_T + \delta_S) \cdot p_4(t) + (\delta_C + \delta_S) \cdot p_5(t) + (\delta_T + \delta_S) \cdot p_6(t) + \delta_S \cdot p_7(t) + \delta_S \cdot p_8(t) + \delta_S \cdot p_9(t)$$

This whitepaper is not trying to make a case that standard, non-safety rated, fault detection mechanisms in the Black Channel can formally be used as safety diagnostics of a safety function, since they are not designed to be conformant to IEC 61508. Rather, what this whitepaper conveys is that when modelling the bit error probability of a communication channel, the real-world bit error probability as seen by the safety communication channel (SCL) is very much affected in a positive way by the integrity checkers of the lower, standard communication layers - in particular the Ethernet CRC and the Security HMAC verifications. There are questions as to whether faults such as diffusion errors during security encryption/decryption, or other systematic faults such as buffer overflows will legitimately raise the bit error probability on the wire as seen by the SCL above what IEC 61784-3 specifies as Pe=10^{-2}. However, when analyzing the complete Markov Model, one needs to consider the total system holistically and to view the complete model mathematically. In other words, one should not only consider errors in the Black Channel at the expense of ignoring the inherent Black Channel integrity measures. For example, it would be incomplete to raise the bit error probability, Pe, as seen by the SCL because of security related diffusion or buffer overflow errors at the expense of ignoring the detection capabilities of the Ethernet CRC and Security HMAC and the probabilities that those checkers fail and become void.

Conclusion

In conclusion, we have applied a Markov model to analyze the impact on the overall residual error rate of a combined safety and security system. Based on this analysis, we propose the following:

(1) The security subsystem remains independent of the safety sub-system and does not influence the performance of the safety subsystem, though it will decrease system errors and reinforce the performance of the interface between safety and security. The security layer therefore enhances the performance of the system by checking data integrity before packets reach the point at which they are checked by the safety function
(2) Selection of a Pe value of 0.5 for modelling the bit error probability without considering decoder and interface failure rates can be seen as overly conservative and not practical in a real-world environment
(3) The designers of products should pay attention to the design of the interface as the interface between safety and security, as this has the most impact.

There is however strong evidence that this last point is already being addressed. Safety products need to follow the processes defined in the IEC61508 standard. Similarly, secure devices need to be built according to the processes defined in the IEC62443 standard. Both of these standards encourage the development of products according to robust development processes that aim to minimize defects arising from systemic errors.

References

[1] HMS, https://www.hms-networks.com/news-and-insights/news-from-hms/2021/03/31/continued-growth-for-industrial-networks-despite-pandemic [Accessed 2 Dec 2021]
[2] IEC TS 63074, https://webstore.iec.ch/publication/31572
[3] IEC 61508-4, https://webstore.iec.ch/publication/5518
[4] ODVA Technology Overview: CIP Safety (PUB00110R4)
[5] IEC 61784-3, https://webstore.iec.ch/publication/62095
[6] M. Bellare, T. Kohno, “Hash Function Balance and Its Impact on Birthday Attacks”, in Cachin C., Camenisch J.L. (eds) Advances in Cryptology - EUROCRYPT 2004. EUROCRYPT 2004. Lecture

Notes in Computer Science, vol 3027. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24676-3_24

[7] Shannon, Claude “Communication Theory of Secrecy Systems” in Bell System Technical Journal. Vol 28, October 1949 http://pages.cs.wisc.edu/~rist/642-spring-2014/shannon-secrecy.pdf

[8] M. Franekova, P. Luley, T. Ondrasina, “Modelling of Failures Effect of Open Transmission System for Safety Critical Applications with the Intention of Safety”, in *Elektronika IR Elektrotechnika, ISSN 1392-1215*, vol. 20, no. 1, 2014.

****

The ideas, opinions, and recommendations expressed herein are intended to describe concepts of the author(s) for the possible use of ODVA technologies and do not reflect the ideas, opinions, and recommendation of ODVA per se. Because ODVA technologies may be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the reader and those responsible for specifying ODVA networks must determine for themselves the suitability and the suitability of ideas, opinions, and recommendations expressed herein for intended use.