Cyberdéfense des infrastructures critiques

Stéphane Mocanu

Cyberdéfense des infrastructures critiques

Mémoire présenté en vue de l'obtention de l'Habilitation à Diriger les Recherches

Stéphane Mocanu

U.G.A, CNRS, G-INP, Inria, LIG

16 janvier 2019

Jury :

Rapporteurs :

Ludovic Mé, CentraleSupélec
Eric Rondeau, CRAN
Yves Le Traon, Université de Luxembourg

Examinateurs :

Philippe Dhaussy, ENSTA Bretagne
Marie-Laure Potet, Grenoble-INP
Eric Ruten, INRIA

Préambule

Ce manuscrit présente mes recherches dans le domaine de la cybersécurité des systèmes industriels développées depuis 2013, ainsi que les perspectives pour les années à venir. Ces recherches ont été réalisées au sein du GIPSA-lab (2013-2016) et du Laboratoire d'Informatique de Grenoble (à partir de 2016). La thématique principale est la détection des intrusions cyber physiques (ou « orientées processus ») avec des contributions dans le monitoring des propriétés de sécurité physiques, la normalisation des alertes multi-domaine et la corrélation des alertes.

La présentation des contributions est partagée en deux volets applicatifs : systèmes séquentiels et systèmes d'automatisme des postes électriques. La distinction est justifiée par les particularités des deux domaines dont la conséquence est l'utilisation de modèles et méthodologies spécifiques.

Deux thématiques connexes : la recherche des vulnérabilités par rétro-ingénierie et l'approche autonomique pour la cyberdéfense des infrastructures critiques sont des thèmes que j'ai commencé à développer récemment. Le chapitre concernant ces deux thématiques se trouve dans la partie dédiée aux perspectives de recherche.

Enfin, un chapitre est dédié à la plate-forme expérimentale, aux techniques de simulation « matériel dans la boucle » et aux mesures de performance des réseaux temps-réel. Ces aspects représentent une partie non-négligeable de mon activité.

La cybersécurité n'a pas toujours été ma thématique de recherche. Après avoir fait une thèse en systèmes stochastiques, j'ai encadré trois thèses sur l'évaluation des performances et commande des systèmes stochastiques. Bien que l'activité cybersécurité soit largement dominante ces six dernières années, je maintiens une activité dans les systèmes stochastiques. Un dernier chapitre présente la synthèse de ces travaux.

Mes recherches dans la cybersécurité des SCADA se situent à la frontière de l'informatique avec deux autres domaines : l'automatique et le génie électrique. L'exercice de présentation est délicat car le document se veut compréhensible par les chercheurs des trois communautés. Les notions basiques de l'automatique, des réseaux électriques et de la détection des intrusions sont reprises dans le manuscrit afin qu'il puisse être présenté dans un format auto contenu.

Table des matières

1 Introduction

Les *systèmes automatisés de contrôle des procédés* constituent un cas particulier de systèmes informatiques. Tout au long de ce document nous allons nous appuyer sur la définition de l'ANSSI [1] qui les désigne comme *« un ensemble de moyens humains et matériels ayant pour finalité de contrôler ou commander des installations techniques (composées d'un ensemble de capteurs et d'actionneurs) »*. Issus de l'informatisation des applications de l'automatique, ces systèmes héritent des dénominations décrivant leurs caractéristiques. Ainsi, selon le contexte de l'application industrielle on parle de systèmes SCADA (Supervisory Control And Data Acquisition) pour des systèmes de commande hiérarchiques et plutôt centralisés, DCS (Distributed Control System) pour des systèmes plutôt distribuées ou PCS (Process Control System) pour les systèmes dont le comportement est dominé par des dynamiques continues.

Dans le contexte de l'étude, la distinction entre SCADA, DCS et PCS n'est pas importante. On utilisera plutôt les termes génériques adoptés par la plus grande partie de la littérature : IACS (Industrial Automation and Control Systems), ICS (Industrial Control Systems) ou tout simplement systèmes industriels ou infrastructures critiques.

Le processus physique exact n'est pas important pour l'approche générale. Aujourd'hui la tendance est d'inclure dans les infrastructures critiques tous les systèmes dont les attaques peuvent avoir comme conséquences les atteintes à l'intégrités des personnes, du matériel et de l'environnement. Cela inclut, au-delà des sites industriels classiques, les systèmes de transport, la GTB, les systèmes médicaux, etc.

Notons que l'automatisme classique distingue deux classes de systèmes industriels : les systèmes dits « manufacturiers », dont les variables sont discrètes et les « procédés », dont les variables sont continues.

Notre étude prend en compte les systèmes manufacturiers ainsi qu'une classe particulière des procédés.

1.1 Architecture des systèmes industriels.

Nous allons adopter l'architecture de référence proposée par ANSSI [2]. La spécification est basée sur le modèle Computer Integrated Manufacturing [3] et permet de décrire la complexité et les fonctionnalités d'un système industriel. Notons que des modèles similaires ont été proposés par la NIST [4] et par la CEI [5].

L'architecture (Figure 1.1) identifie quatre niveaux (catégories d'équipements) avec des particularités de communication spécifiques :

- CIM 0 (niveau terrain) : capteurs, actionneurs, E/S/ déportées. Communication temps-réel[1].
- CIM 1 (niveau contrôle) : automates programmables industriels, analyseurs, pupitres opérateurs, stations d'ingénierie. Communication temps réel faible (cadencement sur temps de cycle automate).
- CIM 2 (niveau supervision) : supervision et historisation locale. Communication avec des contraintes temps-réel très faibles ou pas de temps-réel.
- CIM 3 (niveau gestion) : supervision distante, journalisation globale, planification de la production, ERP. Communication sans contraintes temps-réel.

En s'appuyant sur les quatre couches identifiées, trois niveaux de complexité (fonctionnalité) ont été définis : simple (CIM 0 et 1) complexe (CIM 0, 1 et 2) et très complexe (tous les niveau CIM).

[1] Dans cette étude la notion de temps-réel est considérée dans le sens déterministe de la garantie de la borne du délai de propagation de bout-à-bout. Le temps réel dur implique l'échec de l'application en cas de violation des contraintes, le temps réel faible implique seulement une dégradation de la performance.

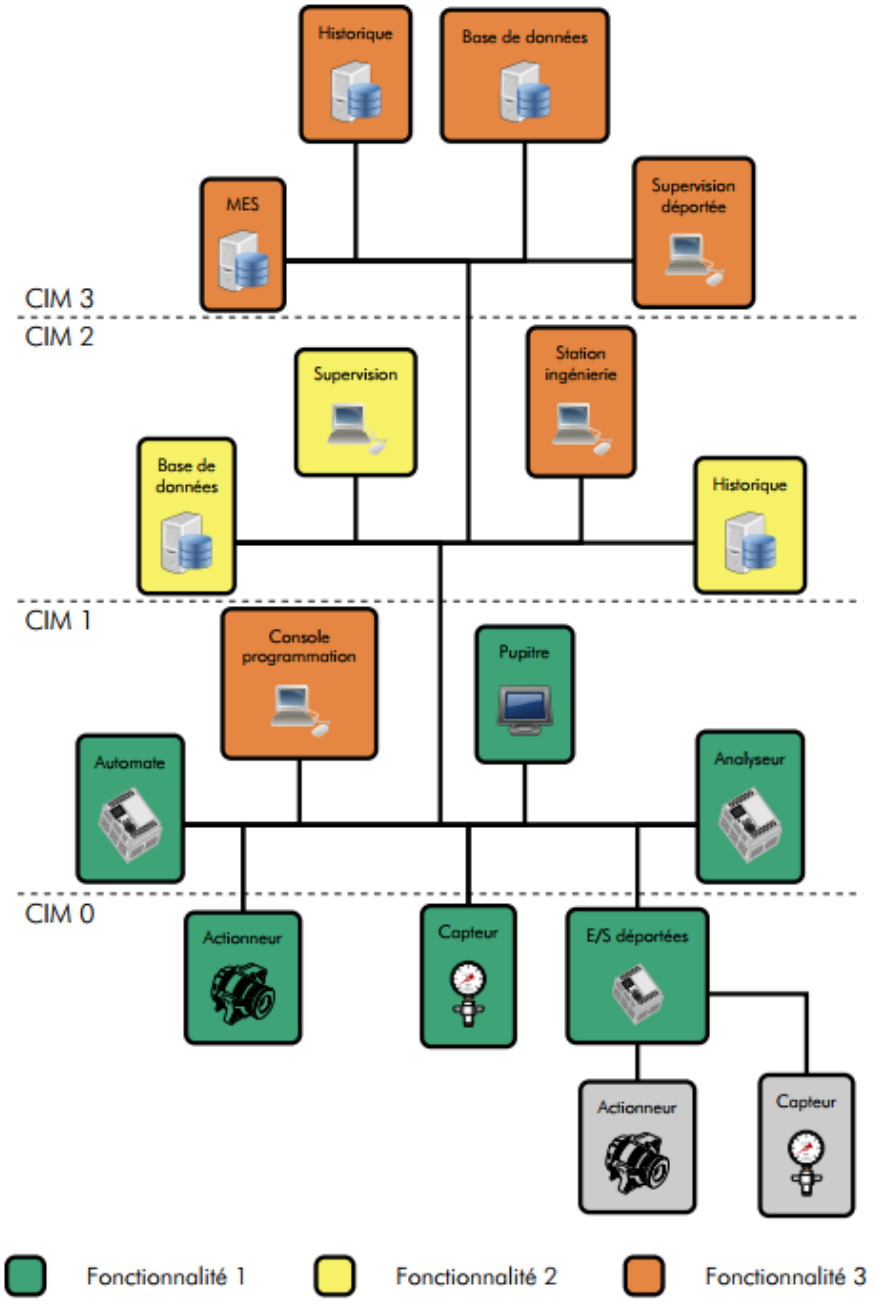


Figure 1.1 Architecture de référence des systèmes industriels

1.1.1 Equipements et logiciels embarqués

Les niveaux CIM 0 et 1 comprennent essentiellement des équipements embarqués spécialisés (capteurs communicants, E/S déportées, automates programmables, relais de protection, écrans de supervision) alors que les niveaux CIM 2 et 3 sont plus proches des systèmes informatiques classiques, leurs composantes étant essentiellement des logiciels client et serveurs s'exécutant sur des ordinateurs standard.

Quelques caractéristiques des équipements présents au niveaux terrain et contrôle sont importantes pour l'analyse de la sécurité. Nous allons analyser principalement le cas des automates programmables dont les caractéristiques couvrent celle des autres équipements.

- **Modularité.** Un automate programmable est « construit » autour d'un module principal (appelé souvent « CPU » par abus de langage) avec des modules d'extension (entrées/sorties) et des

modules « métier » : modules de communication, modules spécialisés pour le comptage, pesage, commande des moteurs, etc., Les modules communiquent entre eux par un bus électronique interne. Les modules métier sont assez souvent munis de leur propre processeur et système d'exploitation. La conséquence de cette caractéristique est que, en réalité, un automate programmable est un conglomérat de plusieurs processeurs et systèmes d'exploitation potentiellement différents communiquant par plusieurs interfaces réseau de nature différente. Couramment, sur un automate moderne on rencontre au moins deux processeurs et systèmes d'exploitation (pour le module principal et le module réseau) et trois interfaces réseau (programmation, terrain et supervision), communiquant en général avec des protocoles et supports physiques différents.

- **Durée de vie.** La durée de vie préconisée d'un automate programmable est d'environ 20 – 30 ans. Pour les relais de protection électrique, le temps de bon fonctionnement préconisé est d'un siècle. Dans certains secteurs industriels (nucléaire par exemple) où la validation des systèmes d'automatisme est très longue est très couteuse, les systèmes des années 80 sont toujours en service. Cela signifie, d'une part, que des systèmes d'exploitation temps-réel anciens non-maintenus sont toujours en service et d'autre part, que les stations d'ingénierie des années 80 avec les environnements de programmation, sont encore présentes dans les systèmes d'information. Même les générations récentes d'automates programmables ont hérité parfois d'une base logicielle ancienne. A titre d'exemple des systèmes d'exploitation embarqués comme pSoS ou basés sur des versions temps réel de MS-DOS (RT-DOS) sont courants sur les équipements de contrôle-commande modernes. Des stations d'ingénieries tournant sous Windows 3.11 ou OS/2 sont toujours en fonctionnement pour la programmation des certains automates en service.
- **Ressources limitées.** Malgré les progrès de l'industrie des circuits intégrés dans les dernières décennies, les ressources disponibles sur les équipements industriels restent très faibles en termes de vitesse de processeur, mémoire vive ou débit réseau. Aujourd'hui (2018) un processeur rapide d'un automate programmable tourne à 400 MHz, quelques méga-octets de RAM sont en général disponibles, les capacités de stockage sont quasi-nulles et FastEthernet est, de facto, le standard industriel de communication. Ces limitations ont deux raisons principales : les difficultés de refroidissement à l'intérieur de boitiers et les contraintes de compatibilité électromagnétiques imposées par les environnements industriels (les postes électriques de transformation, par exemple). Les conséquences directes de ces limitations sont :
    - Une charge importante des processeurs. D'un point de vue cybersécurité, cela limite le déploiement des composantes logicielles de défense au niveau des équipements
    - L'impossibilité de stocker des historiques d'évènements en raison de la (quasi-)absence de capacité de stockage
    - L'impossibilité de chiffrer les flux réseau tout en respectant les contraintes temps-réel

  Les toutes dernières générations d'équipements arrivent à s'affranchir de ces limitations en utilisant des processeurs programmés sur des FPGA à faible consommation et haut débit de calcul. Ainsi, quelques protocoles sécurisés ont été déployés sur des solutions commerciales, quelques cartes GigabitEthernet existent sur le marché et la possibilité d'utiliser de serveurs syslog pour le stockage des évènements est disponible sur certains équipements. Pour l'instant il s'agit de solutions propriétaires et nous ne disposons pas encore des solutions interopérables.
- **Programmation.** Si l'on cherche dans la littérature la définition d'un automate programmable dans les années 80, on trouve : « ordinateur modulaire, durci pour l'environnement industriel programmable par *du personnel non-informaticien* ». En effet le code embarqué sur les équipements de contrôle-commande industriels est généré automatiquement à partir des langages graphiques de programmation plus proches de la spécification que de l'implémentation. Notre étude étant influencée par le langage de programmation, nous allons détailler cette partie.

Le fonctionnement général du programme de l'automate suit, en général, le cycle décrit en Figure 1.2. La tâche principale, généralement activée avec une périodicité fixe, correspond au fonctionnement générique d'un contrôleur : lecture des entrées, calcul des commandes, écriture des commandes.

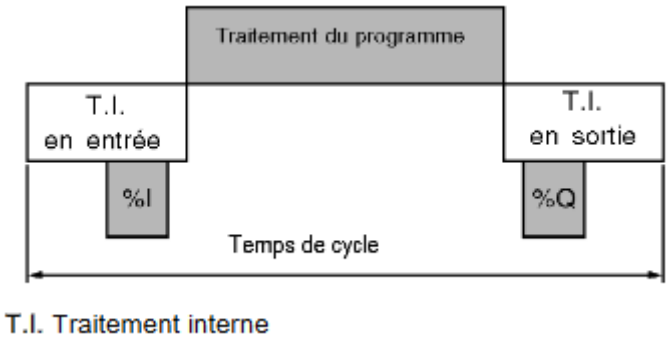


Figure 1.2 Temps de cycle du processeur.

Le logiciel de commande embarqué inclut les parties continues (régulateurs et contrôleurs continus) et une partie discrète (expressions logiques) qui décrit le comportement général du système. Cette partie discrète qui relie les variables booléennes d'entrée et de sortie du système est appelée partie séquentielle. Pour la spécification comportementale de cette partie séquentielle, un langage graphique appelée GRAFCET (GRAphe Fonctionnel de Commande Etape Transition) a été normalisé [6]. Notons que dans les variables booléennes nous incluons, par extension, l'évaluation des prédicats définis sur des variables non-booléennes (test de niveau dans la Figure 1.3) ou l'affectation des variables numériques (consigne du régulateur PID en Figure 1.3).

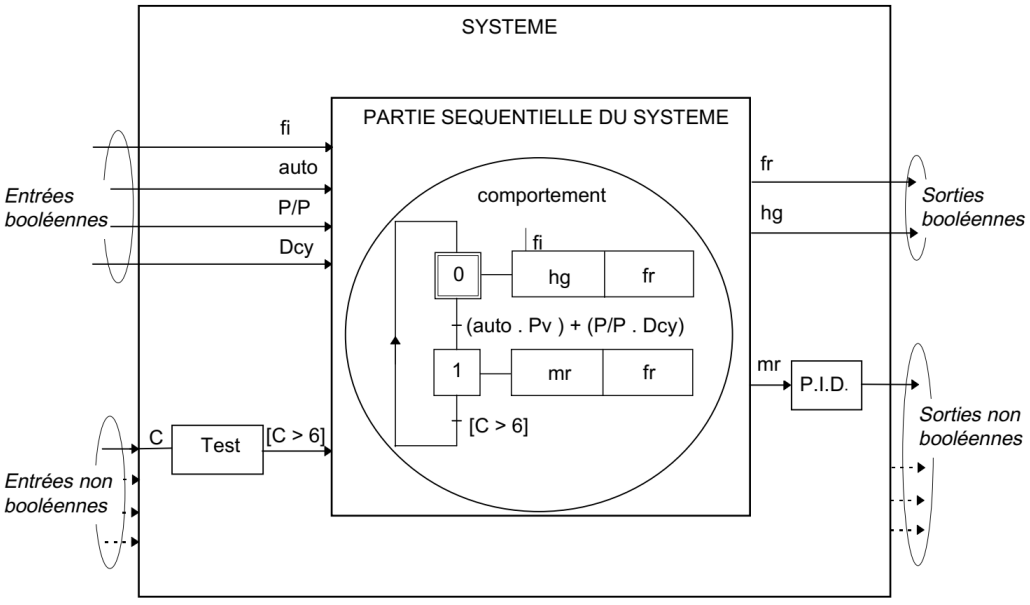


Figure 1.3 Représentation en GRAFCET de la partie séquentielle du système ([6])

Le langage GRAFCET présente de caractéristiques communes avec les réseaux de Petri dont il est issu en partie [7]. Nous distinguons deux parties :

- Une partie de structure, qui est un graphe orienté transition/étape avec les étapes (états) « monomarquées ».

- Une partie interprétation, qui comprend la réceptivité des transitions (condition de franchissement d'une transition) et les actions sur les variables de sortie en mode continu (assignation sur état) ou mémorisé (affectation sur évènement).

La Figure 1.4 illustre les principales notions liées au langage GRAFCET et le lien avec la partie séquentielle du système.

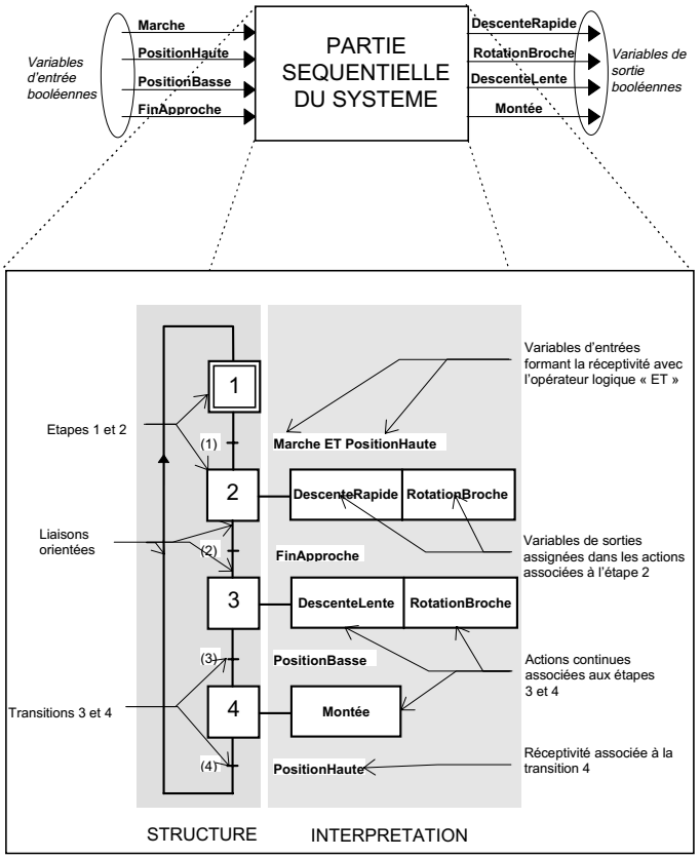


Figure 1.4: Exemple d'utilisation du GRAFCET pour la description comportementale d'un système séquentiel. ([6])

La norme CEI 61131-3 spécifie cinq langages de programmation des automates programmables industriels. Un des cinq langages (Sequential Function Chart – SFC) est censé être utilisé uniquement pour l'organisation interne des programmes de l'automate. Il correspond à l'implémentation de la partie structurelle du GRAFCET. Les deux autres langages graphiques (Ladder Diagram – LD et Function Block Diagram – FBD) correspondent à l'implémentation de la partie interprétation du GRAFCET. Finalement les deux langages textuels (Structured Text – ST et Instruction Logic – IL) peuvent être utilisés pour l'implémentation complète du GRAFCET (partie structure et partie interprétation).

La même norme spécifie la structure des applications embarquées construite avec trois types d'Unités d'organisation de programme (Program Organisation Unit – POU) : programmes, fonctions et blocs fonctionnels. Des bibliothèques standard de fonctions et blocs fonctionnels appelable dans langages graphiques et textuels sont également définies. La dernière version de la norme (2013) introduit des notions orientés objets (classes, interfaces, polymorphisme) mais les environnements de développement industriels actuels ne les prennent pas en charge.

Notons que les deux langages graphiques d'interprétation du GRAFCET (LD et FDB) interdisent les boucles (une variable est évaluée une seule fois par cycle automate), ce qui limite leur utilisation pour les processus continus. Une extension du langage FDB permettant les boucles et l'évaluation continue des variables appelée Graphe de Fonctions Continues (Continous Function Chart – CFC) commence à être de plus en plus utilisée, bien que non

normalisée. C'est notamment le cas des équipements utilisés dans les systèmes de protection des réseaux électriques. Les fonctions de protection complexes ainsi que l'interverrouillage des relais sont programmées en langage CFC.

1.1.2 Protocoles de communication

Une multitude de protocoles de communication ont été spécifiés et déployés au fil du temps. Selon le niveau CIM, la communication doit répondre à des contraintes temps-réel plus ou moins strictes. Ainsi, au niveau CIM 0 (capteurs/actionneurs) une garantie stricte du déterminisme est généralement requise. Certaines communications de niveau CIM 1 (inter-automates) imposent aussi un déterminisme de la communication. A partir du niveau CIM 2 les contraintes temps réel sont très faibles, voire inexistantes.

Chaque constructeur, voire même chaque pays, a tenté de proposer son propre panel de protocoles de communication afin de répondre aux besoins de communication des différents niveaux CIM. Confrontés aux problèmes d'interopérabilité, certains constructeurs ont préféré normaliser leurs solutions par des normes internationales, alors que d'autres ont formé des associations de constructeurs et utilisateurs.

Sans essayer d'être exhaustifs, nous allons rappeler quelques catégories de protocoles et réseaux de communication couramment utilisées dans l'industrie française.

1.1.2.1 Réseaux temps-réel strictes (CIM 0 et 1)

Afin de répondre aux contraintes temps réel strictes, les réseaux Ethernet ont été longtemps bannies des applications industrielles en raison de la présence des collisions et du mécanisme CSMA/CD. Les solutions traditionnelles utilisent des supports physiques simples (RS-485 par exemple) et des mécanismes d'accès assurant le déterminisme : maître/esclave, anneau à jeton, priorité des trames ou ordonnancement total de la communication. Parmi les réseaux implémentant des mécanismes temps-réel stricts sur des supports non-Ethernet, nous pouvons citer les réseaux AS-i [8], les bus de terrain (une vingtaine dont Foundation Fieldbus, Profibus et Hart, parmi les plus connus) spécifiés par les séries de normes CEI 61158/61784-1 [9], [10] ainsi que les réseaux issus de l'industrie automobile tel que CAN [11] ou le standard « de facto » Modbus dont les spécifications sont maintenues par la Modbus Organisation [12].

À la suite de la généralisation des réseaux Ethernet commutées et donc de l'élimination des collisions et de l'indéterminisme, les communications sur support Ethernet ont fait leur apparition dans l'industrie et sont supposées remplacer à long terme les bus de terrain. La partie 2 de CEI 61784 [13] spécifie les caractéristiques des réseaux Ethernet temps-réel ainsi que quelques profils (protocoles) de communication industriels (dont EtherCAT, EtherNet/IP et CIP parmi les plus connus).

1.1.2.2 Réseaux temps-réel faible (CIM 1).

Evidemment, certaines solutions temps-réel strict peuvent être utilisées pour des communications temps réel faible. Cependant, les communications de niveau CIM 1 peuvent impliquer des communications sur de longues distances sur des réseaux reliant des dizaines, voire centaines, d'équipements ou la cohabitation avec du trafic de supervision (souvent non-temps-réel). Les solutions basées sur trames Ethernet et adressage IP constituent la tendance actuelle. Le transport TCP a été assez longtemps évité pour les communications temps-réel en raison de son non-déterminisme (reconnexion, slow-start, contrôle des congestions ou encore l'algorithme de Nagle) et certaines solutions préfèrent UDP avec une gestion d'échanges soit client/serveur (EtherNet/IP pour les E/S ou encore Modbus/UDP) soit éditeur/abonné (RTPS). L'utilisation des options des sockets TCP et la possibilité de crypter les flux TCP[2] ont encouragé le déploiement des protocoles basés sur TCP pour la communication temps-réel faible (Modbus/TCP, S7 ou Ethernet/IP sur TCP).

[2] DTLS 1.0 date seulement de 2006 alors que TLS 1.0 existait depuis 1999

Notons qu'en raison de longues durées de vie des équipements et des coûts très onéreux du changement des équipements, des réseaux utilisant des protocoles OSI (routage X.233 et transport X.224, par exemple) existent encore dans des systèmes industriels.

1.1.2.3 Réseaux d'accès distant (CIM 2 et 3).

Des protocoles industriels portés par TCP/IP de niveau CIM 1 peuvent être utilisés pour la communication intersites ou l'accès distant. Cependant, l'accès multiple à un équipement de contrôle commande risque de pénaliser lourdement l'application temps réel[3]. L'accès distant (CIM 3) devrait communiquer uniquement avec un serveur de collecte de données[4]. Les protocoles d'accès distant concernent, donc, plutôt la communication entre un client et un serveur de collecte de données (SCADA) qui à son tour, communique sur plusieurs réseaux multi-protocole. Actuellement la solution qui semble s'imposer est celle de la communication Open Platform Communications[5] (OPC). La version initiale (OPC DA) basée sur une communication DCOM n'était pas sécurisé, alors que la nouvelle version (OPC UA [14]) permet une communication sécurisée soit sur HTTPS/SOAP soit sur un service TCP dédié[6].

Notons que l'accès distant aux infrastructures critiques de grande taille (réseau EDF, par exemple) a longtemps utilisé les réseaux X.25. En 2011 de tels accès étaient encore en production. Bien qu'Orange ait arrêté son dernier commutateur X.25 en mai 2017, il est fort probable que des accès sur X.25 subsistent sur des infrastructures privées dans les systèmes industriels.

1.1.2.4 Réseaux de communication dans les postes électriques

Depuis la fin des années 90, le domaine du transport et de la distribution électrique a commencé à abandonner les protocoles de communication industriels classiques et a développé ses propres protocoles de communication. Ce choix est justifié par deux raisons principales :

1) L'interopérabilité. Les opérateurs des réseaux électriques sont amenés en permanence à échanger des données avec des partenaires, quel que soit leur domaine d'activité (production, transport, distribution). L'existence des modèles de données uniques et des protocoles interopérables est cruciale.
2) La spécificité des applications. Dans le domaine électrique les fonctions de contrôle/commande (le plus souvent liées à la protection électrique) sont normalisées. Les protocoles de communication font partie intégrante de la fonction de protection et sont optimisés dans ce sens.

Deux générations de protocoles sont utilisées actuellement dans les réseaux électriques correspondant à deux séries de normes CEI 60870 [15] et CEI 61850 [16]. Nous allons discuter plus en détail ces protocoles dans le chapitre dédié à la cybersécurité des réseaux électriques (Chapitre 3).

Nous allons conclure cette brève discussion sur les protocoles de communication dans les systèmes industriels avec deux remarques :

1) Dans une installation existante on rencontre un spectre assez large de protocoles différents et de plusieurs générations. Même dans les installations nouvelles on trouve plusieurs protocoles de communication répondant à des besoins temps réel différents.
2) A quelques rares exceptions près, les protocoles de communication ne sont pas sécurisés.

[3] A titre d'exemple sur la plate-forme d'automatisme M580 de Schneider la carte de communication dédiée NOC0301/11 supporte un maximum de 16 connexions client ou serveur. Le module processeur P581020 permet au maximum 16 connexions client et 32 connexions serveur sur la carte Ethernet intégrée pour un maximum de 500 transactions/s.

[4] Malheureusement, beaucoup d'équipements de contrôle/commande sont exposées directement sur Internet.

[5] Anciennement « OLE for Process Control »

[6] La prochaine version prévoit aussi une transmission UDP sécurisée de type éditeur/abonné

1.2 Cybersécurité des systèmes industriels

La cybersécurité des systèmes industriels est une thématique de recherche relativement nouvelle. Longtemps considérés sécurisés par isolation (pas d'accès à distance) et par obscurité (utilisation des protocoles propriétaires non-publiés), les systèmes industriels ont commencé à faire véritablement l'objet des études en cybersécurité après l'attaque Stuxnet médiatisée en 2010. Des attaques et incidents ont été rapportés avant 2010 mais le manque de médiatisation et une certaine réticence de la part des industriels à admettre la vulnérabilité de leurs systèmes, ont limité l'impact sur la recherche. Même le cas d'école du système de traitement des eaux de Maroochy Shire[7] a commencé à être cité des années plus tard, alors même que des rapports pointaient de doigt la vulnérabilité des systèmes industriels dès 2001[8].

Une prise de conscience d'une partie des industriels a eu lieu après le blackout du nord-est des Etats-Unis en 2003. Bien que déclenché de manière accidentelle, le scénario du blackout pouvait aussi correspondre à une attaque informatique complexe car il a été la conséquence d'une injection de fausses mesures dans le système de commande, combinée avec un bug informatique empêchant la remontée des alertes dans le système de supervision. Dès 2006 des études industrielles suggéraient assez timidement l'existence des vulnérabilités des systèmes industriels et la possibilité de leur exploitation pour des menaces dans le contexte du déploiement des technologies Ethernet/TCP/IP dans l'industrie [17].

La volonté politique des gouvernements successifs a encouragé la recherche dans ce domaine. La création quasi-simultanée en 2009 du *National Cybersecurity and Communications Integration Center* aux Etats-Unis et de *l'Agence nationale de la sécurité des systèmes d'information* en France (dont la participation à la recherche est l'une des missions), montre l'importance que les gouvernements accordent à la problématique de la cybersécurité des systèmes informatiques (dont les systèmes industriels font partie). Le cadre législatif évoluant dans tous les pays (Loi de programmation militaire 2014-2019 en France, par exemple) des normes de sécurité des systèmes industriels ont également vu le jour et par voie de conséquence, la recherche en cybersécurité des infrastructures critiques a pris de plus en plus d'ampleur ces dernières années.

1.2.1 Particularités de la cybersécurité des systèmes industriels

La mise-en-œuvre de la cybersécurité des systèmes industriels présente quelques particularités qui découlent des caractéristiques des équipements informatiques de contrôle/commande mentionnées en Section 1.1.1 et de la nature du processus physique piloté [4].

Enjeux. Des perturbations dans le fonctionnement des équipements de contrôle commande peuvent engendrer des accidents industriels avec des conséquences lourdes sur l'intégrité des personnes et des biens, sur l'environnement et des pertes de capacité de production.

Objectifs de sécurité. Les objectifs de sécurité sont les mêmes que pour les systèmes d'information classiques (Confidentialité, Intégrité, Disponibilité, Traçabilité, Authentification) mais certains objectifs ont une importance et signification particulière dans le contexte des systèmes industriels. Ainsi la **disponibilité** du système est cruciale. Les interruptions de la fonction de contrôle peuvent engendrer la perte définitive du contrôle sur le processus piloté. Par ailleurs, cela implique que le redémarrage d'un équipement en cas d'intrusion n'est pas acceptable dans les systèmes industriels. Les mesures de cyber-protection doivent assurer la continuité du service même en cas de cyber-attaque [18]. L'**intégrité** des données est définie dans le sens de la garantie de non-altération des mesures de capteurs et des commandes envoyées aux actionneurs. La **confidentialité** en tant que chiffrement des flux ne doit pas pénaliser la performance temps réel du système. De la même manière, **l'authentification** ne doit pas empêcher l'intervention normale des opérateurs.

[7] https://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/
[8] http://www.iwar.org.uk/cip/resources/utilities/SCADAWhitepaperfinal1.pdf

Limitation des ressources. Les équipements industriels ont été conçus pour accomplir les tâches de commande et peuvent ne pas disposer de ressources suffisantes pour les composantes de sécurité (en particulier pour la détection d'intrusions hôte ou le chiffrement des communications).

Le Tableau 1-1 synthétise les principales différences entre la cybersécurité des systèmes informatiques classiques - IT et industriels - OT (adapté de [4]).

Catégorie	IT	OT
Performances requises	• Pas de temps-réel • Cohérence de la réponse • Haut débit • Delai de réponse et gigue acceptables • Peu ou pas d'interaction critique de l'opérateur • Un contrôle d'accès stricte est possible	• Temps-réel • Criticité du temps de réponse • Faible débit • Temps de réponse et gigue acceptables • Interventions d'urgence de l'opérateur • Le contrôle d'accès ne doit pas pénaliser l'interaction avec l'opérateur
Disponibilité	• Redémarrage en cas d'attaque acceptable • Indisponibilités ponctuelles acceptables	• Redémarrage en cas d'attaque non-acceptable • La disponibilité est critique. Systèmes souvent redondants • Arrêts planifiés longtemps à l'avance (semaines/mois) • Temps de validation des modifications très long
Gestion des risques	• Biens à protéger : données • Principaux objectifs de sécurité : confidentialité et intégrité • Tolérance aux fautes moins importante • Risques majeurs : retard des opérations	• Biens à protéger : processus physiques • Principal objectif de sécurité : intégrité des biens et des personnes • La tolérance aux fautes est essentielle • Risques majeurs : mise en danger des personnes, de l'environnement et des biens, non-conformité des produits
Systèmes d'exploitation	• Homogènes et standard • Mises-à-jour faciles et automatisées	• Eclectiques et propriétaires • Mises-à-jour difficiles
Contraintes des ressources	• En général suffisantes pour le déploiement des composantes de sécurité	• Systèmes conçus pour des applications de contrôle/commandes embarquées avec des ressources généralement insuffisantes pour le déploiement des composantes de sécurité
Communication	• Protocoles standard • Généralement réseau principal câble avec quelques composants sans-fil	• Protocoles propriétaires et standard mélangés • Plusieurs types de supports physiques interconnectés • Architectures réseau complexes
Gestion des mises-à-jour	Possible le jour même avec des procédures automatisées	Processus de validation long par rapport à l'intégrité de l'application de contrôle/commande. Planification en avance des mises-à-jour. Certains systèmes ne sont plus maintenus.
Service informatique	Standard	D'habitude constructeur (unique)
Durée de vie	3-5 ans	Dizaines d'années

Localisation	Local informatique	Equipements parfois isolés et difficiles d'accès

Tableau 1-1 : Principales différences entre les systèmes industriels et les systèmes informatiques classique (repris de [4]).

1.2.2 Attaques orientées processus

Nous allons rappeler par la suite quelques caractéristiques de deux des attaques célèbres sur des systèmes industriels, ces caractéristiques ayant motivé notre approche.

1.2.2.1 Stuxnet

Probablement la plus médiatisée des attaques sur des systèmes industriels, Stuxnet est un ver informatique très sophistiqué découvert en 2010. Sa charge est très spécialisée et on suppose qu'elle a été conçue pour endommager les installations nucléaires iraniennes, en particulier l'usine d'enrichissement d'uranium de Natanz [19]. Dans le contexte de l'étude présente nous allons nous limiter à l'analyse de l'attaque, bien que les mécanismes de propagation, de mise-à-jour et de contournement des anti-virus soient aussi très complexes et intéressants.

L'objectif de Stuxnet étant d'endommager physiquement des centrifugeuses, le moyen choisi a été de forcer les rotors des centrifugeuses de tourner à des régimes normalement interdits (trop vite ou trop lentement). Les moteurs des centrifugeuses du système ciblé étant commandés par des variateurs de fréquence pilotés à travers un bus Profibus par des automates Siemens S7-300 supervisées par un SCADA Siemens WinCC, la charge a été conçue pour attaquer spécifiquement ces équipements. En phase d'attaque finale, le ver devait en même temps envoyer les commandes malveillantes aux variateurs de fréquence et fausser la vue du SCADA afin de cacher ses actions.

Afin d'envoyer les commandes malveillantes, Stuxnet devait prendre le contrôle sur l'automate de commande. Fausser la vue du SCADA nécessitait soit la prise de contrôle sur le SCADA, soit la corruption des variables lues par le SCADA dans la mémoire de l'automate. La deuxième solution a été choisie pour Stuxnet.

Déroulement de l'attaque une fois le réseau d'entreprise infiltré :

1) Recherche des stations d'ingénierie permettant la programmation des automates via l'environnement Step 7
2) Corruption de la bibliothèque de communication de Step 7
3) Chargement du code malveillant dans l'automate S7-300 par détournement des requêtes légitimes de l'environnement S7
4) Sur l'automate : reconnaissance de la présence des variateurs ciblés sur bus Profibus
5) Enregistrement des séquences normales des valeurs des capteurs du processus
6) Exécution des séquences malveillantes de commandes des variateurs et rejeu des séquences normales des capteurs vers le SCADA WinCC

Deux remarques concernant ce déroulement d'attaque sont importantes et mettent en évidence l'aspect « orienté processus ».

1) La mise en œuvre a nécessité une connaissance profonde du processus physique et du système de contrôle/commande. Autrement dit, la réalisation du code Stuxnet a nécessité des connaissances en informatique ET automatique. Quelques études et certains détails trouvés dans le code de Stuxnet montrent une connaissance complète de l'installation et des systèmes de commande et sécurité [20].
2) Les communications réseau durant les six étapes précédemment citées ont été parfaitement « légales ». Il n'y a pas eu des violations de la syntaxe ou de la sémantique des protocoles. Pas de requête inconnue et, évidemment, pas de signature. Toutes les commandes envoyées étaient légales. Aucun IDS de l'époque n'aurait levé d'alerte. Aucune anomalie n'aurait pu être relevée au niveau du réseau. La seule possibilité de détecter une anomalie aurait été de comparer les

valeurs des capteurs lus sur bus Profibus avec les valeurs vues par le SCADA, ce qui suppose l'utilisation de la connaissance du processus physique dans la détection.

1.2.2.2 Industroyer/CrashOverride

Industroyer [21] [22] est le dernier malware en date spécialisé dans les systèmes industriels. Il a été utilisé lors des attaques sur le réseau électrique ukrainien en décembre 2016. Contrairement à Stuxnet, il n'utilise pas une connaissance précise de l'architecture de la cible, mais il utilise des mécanismes de construction d'une cartographie du système attaqué. Il est spécialisé dans les réseaux de distribution électrique. Ces systèmes présentent deux particularités par rapport au déploiement (ces points seront repris et détaillés dans le Chapitre 3) :

- D'une part les fonctions de commande sont standardisées quel que soit le constructeur des équipements. Cela signifie qu'une reconnaissance préalable sur place n'est pas nécessaire, contrairement au cas des installations nucléaires iraniennes.
- D'une autre part une connaissance de la répartition des charges électriques dans le réseau est nécessaire, afin que les actions malveillantes (ouverture d'un disjoncteur isolant un tronçon électrique) aient un impact maximal sur l'ensemble du réseau.

Industroyer hérite des caractéristiques de BlackEnergy 1, 2 et 3 [23], les malwares utilisés lors des attaques précédents sur le réseau ukrainien. Industroyer est un malware multi-protocole et multi réseau (une partie des attaques ont été perpétrés à travers des bus série). La séquence supposée de déroulement de l'attaque une fois le réseau d'entreprise infiltré, est la suivante :

1) Compromission d'un serveur de supervision OPC et utilisation du serveur OPC pour cartographier le réseau
2) Compromission d'une partie des interfaces opérateur identifiées et ouverture de backdoors
3) Ouverture des connexions avec le centre de contrôle-commande du malware, transmissions des informations recueillies et chargement des nouvelles configurations. Il est possible que des interventions manuelles des attaquants aient eu lieu via les backdoors.
4) Arrêt des processus client légitimes des serveurs de supervision et interfaces opérateur et lancement des programmes malveillants
5) Envoi des commandes aux équipements de protection du réseau, ayant pour résultat l'arrêt de 1/5ème du réseau électrique de la ville de Kiev.
6) Lancement des programmes d'effacement des données et des configurations.

Les caractéristiques suivantes de l'attaque sont importantes pour la motivation de notre approche :

1) Les équipements industriels (les serveurs de supervision dans le cas présent) peuvent être utilisés pour mener des opérations de reconnaissance sans que cela viole la politique de sécurité.
2) La menace a pu se substituer aux programmes de supervision légitimes sur réseau Ethernet mais aussi bus de terrain
3) Les commandes envoyées aux équipements de protection ne violaient pas la syntaxe ou la sémantique des protocoles, comme dans le cas Stuxnet.

En synthétisant les conclusions des deux exemples présentés nous pouvons conclure que :

- N'importe quel équipement du système industriel peut être utilisé pour déclencher les actions malveillantes (automate programmable dans le cas de Stuxnet, superviseurs dans le cas Industroyer)
- La menace peut avoir une connaissance détaillée « à priori » du processus physique et du système de supervision contrôle commande (Stuxnet) ou le construire par une reconnaissance (Industroyer)
- Les attaques peuvent être multi-protocole et multi-medium
- Il est parfaitement possible de fausser la vue des opérateurs sur l'état du processus

- Les trames envoyées ne violaient pas les protocoles de communication.
- Dans le deux cas le caractère malveillant des commandes envoyées aux équipements de terrain était relatif à l'état du processus physique.

La dernière remarque est particulièrement importante pour la détection des intrusions dans les systèmes industriels. En effet, décider si une commande envoyée au processus est malveillante ou pas nécessite une corrélation du trafic réseau avec l'état du processus piloté.

1.2.3 Positionnement des travaux

Parmi les nombreuses thématiques de la cybersécurité des systèmes industriels, nos travaux sont positionnés principalement autour de la détection des intrusions réseau, de la recherche des vulnérabilités des protocoles et de la résilience des architectures.

Le paradigme influençant ce positionnement est l'approche par la modélisation des systèmes cyber-physiques. Nous ne nous intéressons pas à la modélisation et la sécurisation d'un équipement individuel mais à l'architecture du système. Cette approche présente l'avantage de prendre en compte l'intégralité du système industriel (processus physique, équipements numériques, réseaux de communication, intervention des opérateurs).

2 Détection d'intrusions dans les systèmes industriels

Ce chapitre décrit essentiellement les résultats obtenus dans la thèse d'Oualid Koucham, thèse co-encadrée avec Guillaume Hiet (CentraleSupelec) et Jean-Marc Thiriet (Gipsa-Lab), suivis par la présentation des futures directions de recherche. Après un bref rappel des notions basiques sur la détection des intrusions, nous présentons l'approche générale et les principaux résultats.

2.1 Détection des intrusions.

Quels que soient les moyens déployés pour assurer la sécurité d'un système, le risque qu'une intrusion se produise existe car les systèmes d'information sont sujets aux vulnérabilités. La détection d'intrusions intervenant après l'occurrence d'une intrusion, c'est donc une mesure de sécurité a posteriori. L'objectif d'un système de détection d'intrusions (IDS, pour Intrusion Detection System) est d'identifier automatiquement les violations de la politique de sécurité en s'appuyant sur des données acquises du système à protéger. Une méthode de détection permet de décider de la présence d'une intrusion et lever une alerte le cas échéant.

La taxonomie proposée en [24] définit plusieurs caractéristiques permettant la classification des IDS, dont deux sont importantes pour notre étude.

1) **Sources de données**. Nous distinguons les IDS utilisant les données des applications et systèmes d'exploitation des équipements terminaux (Host Intrusion Detection System – HIDS) et ceux basés sur les données réseau, champs des protocoles et données (Network Intrusion Detection System - NIDS).
2) **Méthode de détection**. Deux grandes approches de détection existent. L'approche **comportementale** utilise une description du comportement normal du système afin de détecter les déviations. Le modèle du comportement normal est soit fourni par une spécification « à priori », soit appris par des méthodes statistiques. A l'opposé, l'approche par **connaissances** utilise une description des attaques le plus souvent sous forme de séquence d'évènements ou motifs de données (signatures).

Chaque méthode de détection présente ses avantages et inconvénients. Intuitivement on comprend qu'une approche par signatures ne pourra pas détecter de nouvelles attaques, alors que l'approche comportementale pourra les détecter. D'une autre part, si le modèle du comportement normal n'est pas parfait, un IDS comportemental risque de lever beaucoup de fausses alertes. Notons que la plus grande partie de IDS du commerce sont basés sur des signatures.

Evaluation des performances. En fin de compte un IDS est un classificateur binaire. A chaque décision quatre résultats sont possible :

1) vrais négatifs (True Negatives - TN),
2) vrais positifs (True Positives - TP),
3) faux positifs (False Positives - FP),
4) faux négatifs (False Negatives - FN).

Les diagnostics corrects d'un IDS correspondent à la détection d'une vraie intrusion (TP) ou reconnaissance d'une situation normale en tant que telle (TN), alors que les erreurs de diagnostic correspondent aux fausses alertes (FP) ou intrusions non-détectées (FN).

La validité intrinsèque de la détection est mesurée par le couple sensibilité/spécificité. La *sensibilité* mesure le pourcentage de vrais positifs sur le nombre total d'intrusions (TP/(TP+FN)) alors que la *spécificité* mesure le nombre de vrais négatifs sur le nombre total d'évènements normaux (TN/(TN+FP)). Un IDS parfait a une sensibilité et une spécificité égales à 1 (pas de Faux négatifs ni des Faux positifs). Notons que la qualité de la détection est donnée par le couple sensibilité/spécificité et pas par les mesures individuelles. En effet, un IDS avec une grande sensibilité et une petite spécificité est un IDS qui alerte tous les évènements. Inversement un IDS avec une grande spécificité et une petite sensibilité est un IDS qui ne lève jamais d'alerte.

2.2 Notre approche de détection orientée processus.

Nous nous plaçons dans le contexte d'un système industriel tel que défini dans la Section 1.1. Notre choix, que nous pensons réaliste par rapport aux pratiques d'exploitation industrielles, est de construire un modèle du système cyber-physique à partir de l'observation non-intrusive du trafic réseau sans attaques et de déduire les propriétés de sécurité à surveiller.

2.2.1 Cadre général

Notre étude est limitée à la partie séquentielle du système de commande. L'étude de la partie continue fait partie des perspectives de notre travail et sera discuté à la fin du chapitre. Sans perte de généralité nous supposons que tous les programmes embarqués sont spécifiés en GRAFCET et que la partie structurelle est implémentée en SFC[9] (Section 1.1.1). Le langage d'implémentation de la partie interprétée n'a pas d'importance.

Notre IDS est basé sur l'observation passive du trafic réseau à tous les niveaux CIM. Nous supposons que nous pouvons intercepter tout le trafic réseau. Dans la partie expérimentale nous nous sommes limités au trafic Ethernet principalement en raison des limitations des logiciels de capture de trafic actuels (Wireshark). Le choix d'un IDS réseau est motivé principalement par les contraintes de l'industrie : en effe,t les outils de développement de bas niveau (compilateur C ou similaire) ne sont pas distribués par les constructeurs. Il est donc impossible de rajouter des briques de base aux environnements de développement CEI-61131.

2.2.2 Modèle de la menace.

Nous supposons que l'attaquant a une parfaite connaissance du processus industriel et des algorithmes de commande. Il connait l'architecture réseau ou il peut la construire à la suite d'une reconnaissance. Nous nous intéressons aux attaques orientés processus dans l'esprit des exemples cités précédemment (Stuxnet et Industroyer). Plus précisément nous nous intéressons aux attaques dites « de séquence » ([25] [26]). Ce sont des attaques qui violent la logique séquentielle d'exécution des actions par le contrôleur et sont de deux types : qualitatives ou quantitatives. Nous illustrons les deux types d'attaques sur un exemple simple.

[9] Le choix du langage SFC facilite la structuration du système par états mais la même analyse peut être menée pour des programmes en IL ou ST.

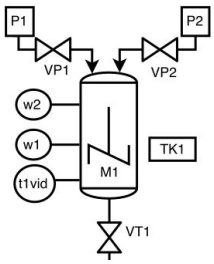


Figure 2.1 Processus de mélange de deux produits.

Exemple 1.. Considérons le processus simple de la Figure 2.1. Il s'agit d'un processus de mélange de deux produits P1 et P2 qui sont mélangés dans le réservoir TK1. Au début de l'opération le réservoir est vide. En ouvrant la vanne VP1 le réservoir est rempli de produit P1 jusqu'à l'activation du capteur de niveau W1. Ensuite la vanne VP1 est fermée et VP2 est ouverte et le produit P2 est rajouté jusqu'à l'activation du capteur W2 quand la vanne VP2 sera fermée. Le moteur M1 est alors démarré pour une période prédéterminée afin de réaliser le mélange. A la fin, le moteur est arrêté et le réservoir est vidé en ouvrant la vanne VT1.

La séquence de commandes réalisant le comportement correct (spécifications) est implémentée dans la logique séquentielle du contrôleur. Les attaques orientées processus vont violer les spécifications, *i.e.* forcer l'occurrence des comportements incorrects. Les attaques qualitatives vont violer la séquence d'évènements. Par exemple, pour le processus considéré, l'ouverture de VP2 avant la fermeture de VP1 et l'activation de W1 modifie les proportions de produits dans le mélange résultant dans un produit non-conforme. Une attaque quantitative vise la temporisation entre les évènements. Par exemple, une séquence rapide d'ouvertures/fermetures des vannes va endommager les actionneurs par usure. Un non-respect de la durée du mélange (arrêt prématuré du moteur) compromettra la qualité du produit final.

Notre approche est orientée surtout vers la détection de attaques qualitatives par une technique de monitoring des propriétés de sécurité du processus physique. Dans l'exemple précédent une propriété de sécurité à surveiller serait « pas d'ouverture de VP2 avant fermeture de VP1 ».

2.2.3 Architecture du système de détection

L'architecture globale du notre système (Figure 2.2) est dérivée de l'architecture générale d'un IDS avec l'ajout de quelques blocs spécialisés.

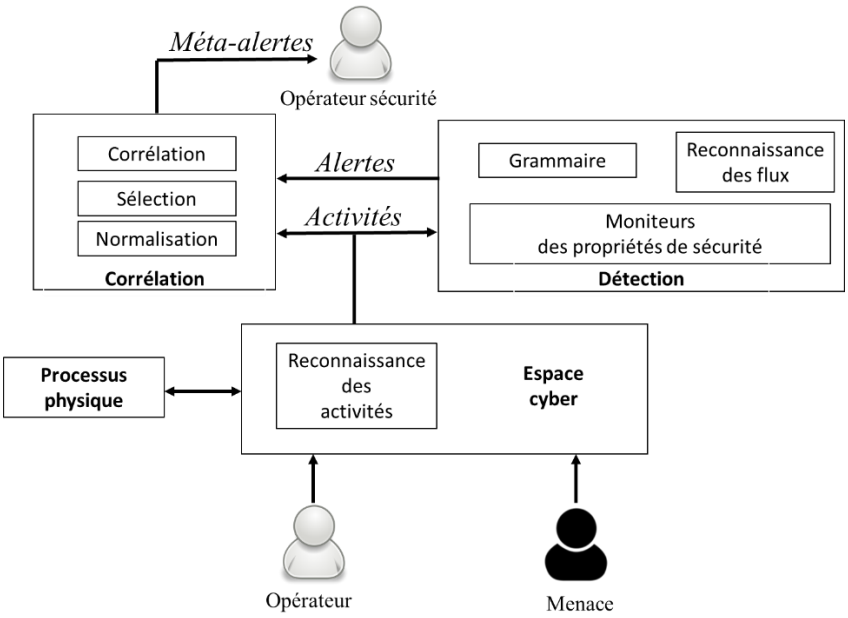


Figure 2.2 Architecture globale du système détection

Ainsi, les propriétés à surveiller étant dépendantes de l'état du système (processus physique et état de la commande) nous devons, idéalement, définir pour chaque état de chaque SFC de chaque automate les propriétés de sécurité correspondantes. En pratique, en se plaçant au niveau de la communication réseau, une partie des états du SFC ne peuvent pas être identifiés. Nous avons défini la notion plus générique d'activité (développée plus loin) qui permet de sélectionner les propriétés de sécurité valides à un moment donné.

Le bloc de détection utilise les observation réseau ainsi que les informations du bloc de reconnaissance des activités pour alimenter plusieurs détecteurs : dans le domaine cyber, un IDS basé sur la grammaire des échanges réseau et un IDS basé sur la reconnaissance des flots réseau et dans le domaine physique, un IDS basé sur la surveillance des propriétés de sécurité.

Finalement, une corrélation entre les alertes des différents IDS permet la réduction du nombre de faux positifs et la reconstitution des scénarios d'attaque.

2.3 Monitoring des spécifications de sécurité

Les spécifications de sécurité sont des propriétés sur des séquences d'évènements dont l'occurrence mène le processus dans un état interdit. Les évènements dans notre système sont relatifs aux capteurs et actionneurs.

La manière évidente d'obtenir les propriétés de sécurité serait de les extraire à partir de la documentation du système de contrôle/commande. En théorie, lors de la synthèse de la commande, les spécifications de sécurité ont été utilisées pour le calcul de l'automate de commande. Les méthodes de synthèse de contrôleurs, par exemple la commande supervisée par l'approche de Ramadge et Wonham [28] utilisent la spécification des états interdits, ainsi qu'un modèle dynamique du processus et la liste des évènements contrôlables et incontrôlables pour obtenir un contrôleur sous la forme d'une liste d'évènements interdits pour chaque état. Ces spécifications sont souvent présentées sous forme d'invariants des réseaux de Petri, automates finies ou formules de logique temporelle.

Nous affirmons qu'une telle approche n'est pas adéquate pour la détection pour plusieurs raisons :

1) la spécification de sécurité utilisée pour la commande est insuffisante pour la détection des intrusions car elle utilise un modèle du processus en absence d'attaques alors que l'attaquant peut forcer le déclenchement des évènements qui n'arriveraient pas dans la dynamique normale du processus
2) dans un système de commande distribuée ces spécifications sont locales pour chaque contrôleur. Il serait nécessaire soit de déployer un IDS pour chaque automate, soit de construire une très compliquée spécification globale par produit parallèle des spécifications locales
3) finalement, dans la vie réelle, en raison du grand nombre de sous-traitants intervenant dans les systèmes d'automatismes industriels et de la longue durée de vie des installations, obtenir les spécifications de sécurité d'une installation réelle est une tâche ardue. Souvent il s'agit simplement d'un cahier de charges avec des spécifications textuelles qui ont été prises en compte dans la réalisation de commande sans aucun formalisme.

Nous avons décidé d'apprendre les spécifications de sécurité du système directement à partir d'une trace d'exécution sans attaques. Ceci présente par ailleurs l'avantage d'inclure une image du comportement de l'opérateur. Bien entendu, les spécifications apprises peuvent être enrichies avec des spécifications de sécurité « à priori ».

2.3.1 Monitoring par vérification à l'exécution

Notre objectif étant de vérifier automatiquement la violation des propriétés séquentielles d'un processus physique pendant son exécution et avec un temps de décision court (idéalement avec une garantie de temps de réponse), nous avons fait le choix de la technique de vérification à l'exécution (Runtime vérification [29]).

2.3.2 Choix du langage de spécification

Le choix de la vérification à l'exécution pour la détection des violations des propriétés de sécurité, limite le choix du langage de spécification. L'exécution de la vérification sur des traces finies impose le choix d'un langage avec une sémantique finie. Le langage de spécification doit aussi nous permettre d'exprimer des propriétés temporelles qualitatives et quantitatives. La logique temporelle linéaire (LTL [30]) est un langage déclaratif utilisé pour les spécifications de sécurité, la vérification à l'exécution ([31]) et même pour la synthèse des contrôleurs ([32]).

LTL est une extension de la logique propositionnelle avec des opérateurs temporels permettant ainsi d'exprimer les relations d'ordre. Les formules LTL sont construites avec les opérateurs logiques ¬, ∧, et les opérateurs temporels X (next), et U (until). La formule $X\varphi$ signifie que φ sera vrai dans l'état suivant, tandis que a$U\ b$ signifie que a sera vrai au moins jusqu'à ce que b devienne vrai et b doit devenir vrai à un moment. Les opérateurs logiques supplémentaires ∨ , ⇒,⇐,⇔, *vrai* et *faux* sont définis de manière habituelle. Au moins deux autres opérateurs temporels sont dérivés : G (toujours) et F (finalement). La formule $F\ \varphi = \ faux\ U\ \varphi$ signifie que, à un moment, φ deviendra vrai. Enfin, $G\varphi =$ ¬F¬φ signifie que φ est vrai partout.

Si p est une proposition atomique $p \in\ AP$ la syntaxe de LTL sur l'alphabet $\Sigma = 2^{AP}$ est définie par

$$\varphi :: \ p \ | \neg\varphi \ | \ \varphi \vee \varphi \ | \ \varphi \mathrm{U}\ \varphi \ | \ X\varphi,\ p \ \in AP$$

La sémantique de LTL est définie de la manière suivante. Soit $w \in \Sigma^{\infty}$ un mot de l'ensemble des séquences infinies Σ^{∞}. Soit $w(i)$ le $i^{ème}$ élément de la séquence. Alors :

$$\begin{array}{lll} w, i \vDash p \in AP & \Leftrightarrow & p \in w(i) \\ w, i \vDash \neg\varphi & \Leftrightarrow & w, i \nvDash \varphi \\ w, i \vDash \varphi_1 \vee\ \varphi_2 & \Leftrightarrow & w, i \vDash \varphi_1 \vee w, i \vDash \varphi_2 \\ w, i \vDash w, i \vDash \varphi_1 \mathrm{U}\ \varphi_2 & \Leftrightarrow & \exists k \in \mathbb{N}, k \geq i. w, k \vDash\ \varphi_2 \wedge \forall i \leq j < k. w, j \vDash \varphi_1 \\ w, i \vDash X\varphi & \Leftrightarrow & w, i + 1 \vDash \varphi \end{array}$$

Dans le contexte de notre étude nous exprimons des propriétés impliquant des évènements telles que les activations et désactivations des capteurs et des actionneurs. Pour chaque capteur (actionneur), deux évènements sont possibles : activation (passage à VRAI, ou front montant) et désactivation (passage à FAUX ou front descendant). Dans un style emprunté aux notations Grafcet nous allons noter les deux évènements par les symboles ↑respectivement ↓. Ainsi l'évènement VP2↑ dénote l'ouverture de la vanne VP2. De tels évènements peuvent être exprimés en LTL par [33] : $a \uparrow \equiv \neg a\ \wedge Xa$ et $a \downarrow \equiv a\ \wedge X\neg a$.

Sur l'exemple d'attaque de séquence de la Section 192.2.2 (***Exemple 1***), la propriété de sécurité exprimant le fait que les deux vannes VP1 et VP2 ne doivent pas être ouvertes simultanément s'écrit en LTL : $G(\neg(VP1 \wedge VP2))$.

2.3.3 Patrons de spécification

Le formalisme LTL permet d'exprimer les propriétés qualitatives concernant l'ordre des évènements[10]. Cependant le formules LTL tendent à devenir rapidement complexes pour des systèmes de grande taille (si on utilise seulement les opérateurs de base de LTL, même la formule simple écrite dans le paragraphe précédent devient compliquée). Des études (dont l'étude originale de Dwyer [34]) montrent que, en pratique, on peut discerner deux grandes catégories de spécifications : celles qui conditionnent l'occurrence d'une propriété et celles qui imposent un ordre. La Figure 2.3 montre la hiérarchie des patrons identifiée en [34]. La signification des patrons est décrite dans le Tableau 2-1.

[10] La question des propriétés temporelles quantitatives sera traitée à la fin du chapitre.

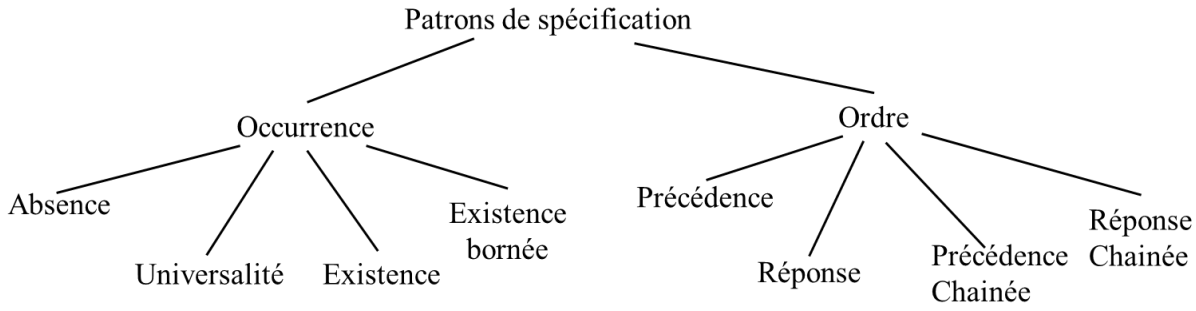


Figure 2.3 Hiérarchie des patrons de spécification

Patron	Signification
Absence	L'état/évènement ne doit pas se produire
Universalité	L'état/évènement est toujours vrai
Existence	L'état/évènement doit se produire
Existence bornée	L'état/évènement doit se produire au minimum/maximum/exactement k fois
Précédence(P,R)	L'état/événement P doit précéder R
Réponse(P,R)	L'état/événement P doit être suivi par R
Précédence chainée	La séquence P1, …, Pn doit précéder la séquence R1,…, Rn
Réponse chainée	La séquence P1, …, Pn doit être suivie par la séquence R1,…, Rn

Tableau 2-1 Types de patrons de spécification

Chaque patron a une portée temporelle définie par rapport à un évènement/état ou par rapport à l'intervalle entre deux évènements/états. Cinq portées temporelles possibles ont été identifiées. Si A et B sont deux évènements/états, les portées possibles sont : *globale, avant A, après B, entre A et B, après A jusqu'à B*. La Figure 2.4 présente graphiquement l'effet de portées des patrons.

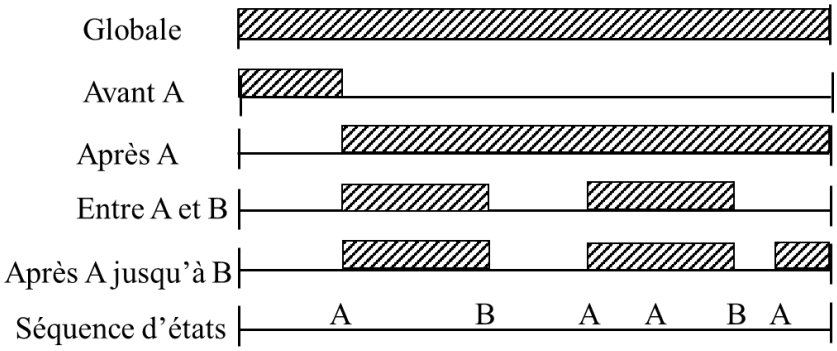


Figure 2.4 Portées possibles des patrons de spécification

En appliquant les patrons de spécification à l'***Exemple 1*** de la Section 2.2.2, la propriété de sécurité interdisant l'ouverture de la vanne VP2 entre l'ouverture et la fermeture de VP1 s'écrit : *absence*(VP1↑,VP1↓, VP2) qui est bien plus compréhensible que l'expression LTL correspondante $G((VP1\uparrow \wedge \neg VP1\downarrow \wedge F\ VP1\downarrow) \Rightarrow (\neg VP2\ U\ VP1\downarrow))$

2.3.4 Monitorabilité

L'idée de la monitorabilité est de pouvoir évaluer sur des traces finies des formules interprétées sur des séquences possiblement infinies. Les formules LTL sont en effet interprétées sur des séquences infinies ce que fait que certaines formules LTL ne sont pas monitorisables. Intuitivement une formule LTL est monitorisable s'il est possible de donner un verdict que la formule est satisfaite sur une séquence finie d'états/évènements. Dans l'ensemble de formules monitorisables, deux sous-ensembles importants ont été identifiés : les formules qui peuvent être violées sur des traces finies (propriétés de sécurité – safety) et les formules qui peuvent être validées sur des traces finies (propriétés de garantie – co-safety). Un

exemple simple de formule de sécurité est G¬p (globalement p ne se produit pas). En effet, toute trace finie qui contient p viole la propriété de sécurité. Une formule de garantie simple est Fp (finalement p se produit). Toute trace finie contenant p valide la formule. Enfin, un exemple de formule non-monitorisable est GFp. Elle ne peut pas être validée sur une trace finie en raison de la présence de l'opérateur G et ne peut pas être violée non plus car cela implique la violation de Fp sur une trace finie, ce qui n'est pas possible. Tous les patrons de spécification ne sont pas monitorisables sur des traces finies.

Nous allons résoudre la question de la monitorisabilité lors de la génération automatique des propriétés de sécurité dans l'étape de fouilles des traces.

2.3.5 Fouille des spécifications

Nous souhaitons donc construire automatiquement l'ensemble des propriétés de sécurité qui seront utilisées dans la phase de détection à partir de l'analyse des traces d'exécution du processus en absence d'attaques. La fouille doit, par ailleurs, nous permettre de mettre en relation les spécifications avec l'état du processus et de limiter le nombre de formules à l'ensemble des spécifications monitorisables (i.e. falsifiables sur une trace finie).

Afin d'atteindre cet objectif nous utilisons dans la fouille : la capture de trafic réseau entre les automates et avec le réseau de supervision (la Figure 2.5 montre un exemple de déploiement des sondes permettant de construire la trace d'exécution) et les programmes SFC des automates à partir desquels nous allons identifier l'état du processus dans la trace.

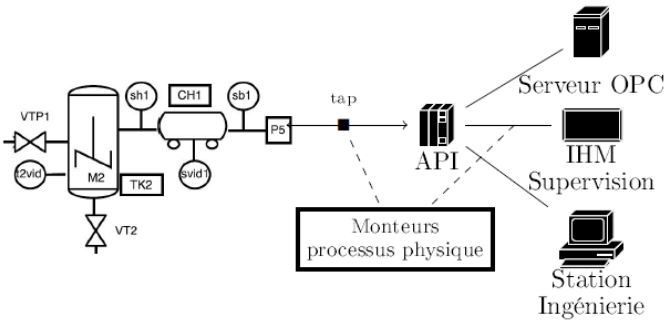


Figure 2.5 Exemple de déploiement des sondes pour la construction des traces d'exécution normale

En nous appuyant sur le programme SFC, nous définissons la notion *d'activité*. La fouille des spécifications sera faite par activité. Pour chaque activité nous identifions l'ensemble des spécifications valides. Afin d'assurer la monitorabilité des spécifications, nous identifions l'intervalle de validité de chaque formule (sa portée). L'ensemble de la démarche est présenté en Figure 2.6 et a fait l'objet d'une publication en [35].

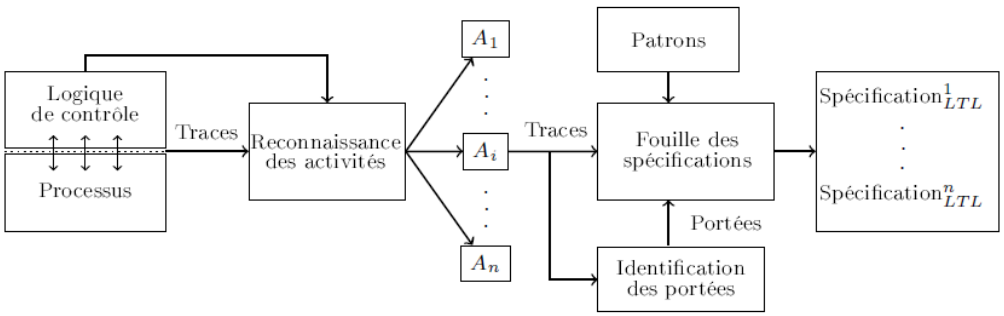


Figure 2.6 Schéma global de la fouille des spécifications

2.3.5.1 Activités

Considérons à nouveau l'***Exemple 1***. De point de vue de la logique du processus nous pouvons identifier quatre opérations : remplissage avec le produit P1, remplissage avec le produit P2, mélange, vidange. Le programme de commande Grafcet réalisant cette séquence est celui de la Figure 2.7.

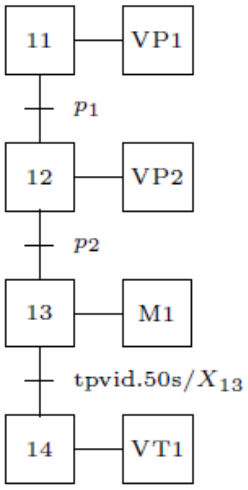


Figure 2.7 Grafcet de commande pour l'Exemple 1

Nous pouvons remarquer que chaque opération du processus correspond à une étape du Grafcet. Ceci est dû à la simplicité du processus considéré, chaque opération se réalisant par l'activation d'un seul actionneur et chaque transition étant conditionnée par un seul capteur. Dans le cas des processus plus complexes, les Grafcet ont une structure compliquée avec exécutions parallèles et branchements, étapes qui s'exécutent périodiquement ou une seule fois, etc. Le programme SFC présenté en Figure 2.8 contient les principales structures de contrôle d'exécution spécifiées par la norme (les étapes M10 et M21 sont des macro-étapes, ou des sous-programmes).

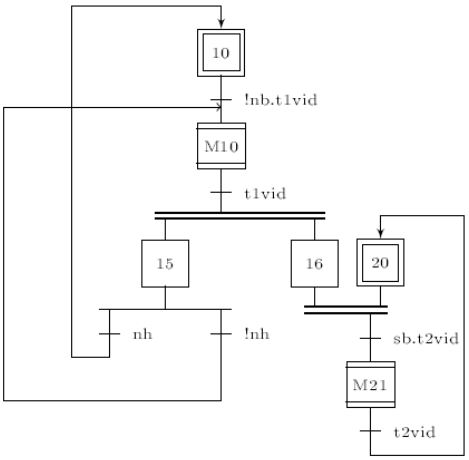


Figure 2.8 SFC complexe avec plusieurs éléments de branchement et contrôle

En général la correspondance entre l'opération technologique et une portion du grafcet n'est pas évidente et nécessite l'utilisation en permanence de la connaissance sur la technologique du processus.

Nous pouvons aussi remarquer que l'individualisation microscopique des opérations technologiques élémentaires dans le SFC n'est pas forcement significative du point de vue sécurité. Par exemple, la condition interdisant que les vannes VP1 et VP2 soient ouvertes simultanément est valable dans les quatre opérations du processus de l'***Exemple 1***. D'autre part, fouiller les spécifications sur l'intégralité de la trace sans tenir compte de l'état du SFC serait une erreur car les propriétés de sécurité différent

selon l'état du processus. Un compromis est nécessaire entre fouiller trop de propriétés en utilisant une vision microscopique (découpage par étape) et rater des propriétés (fouille des traces trop longues).

Nous avons décidé de trouver ce compromis en deux étapes : d'abord on découpe les traces par rapport à des entités que nous appelons « activités », ensuite à l'intérieur de chaque trace nous fouillons les spécifications et chercherons leurs intervalles de validité afin d'assurer la monitorabilité.

Définition 2-1. Une activité est une séquence linéaire d'étapes et transitions qui ne contient pas de structures de contrôle de l'exécution (branchements, exécutions parallèles, synchronisation, etc).

Une activité peut comprendre une ou plusieurs étapes. Par défaut une activité contient toutes les étapes/transitions situées entre deux éléments de contrôle mais si une connaissance « à priori » des propriétés du système l'imposent, elle peut être découpée en plusieurs activités.

Un exemple de découpage en activités du programme SFC de la Figure 2.8 est représenté en Figure 2.9. Le découpage n'est pas unique car les activités M10 et M21 peuvent être découpées en plusieurs activités si besoin.

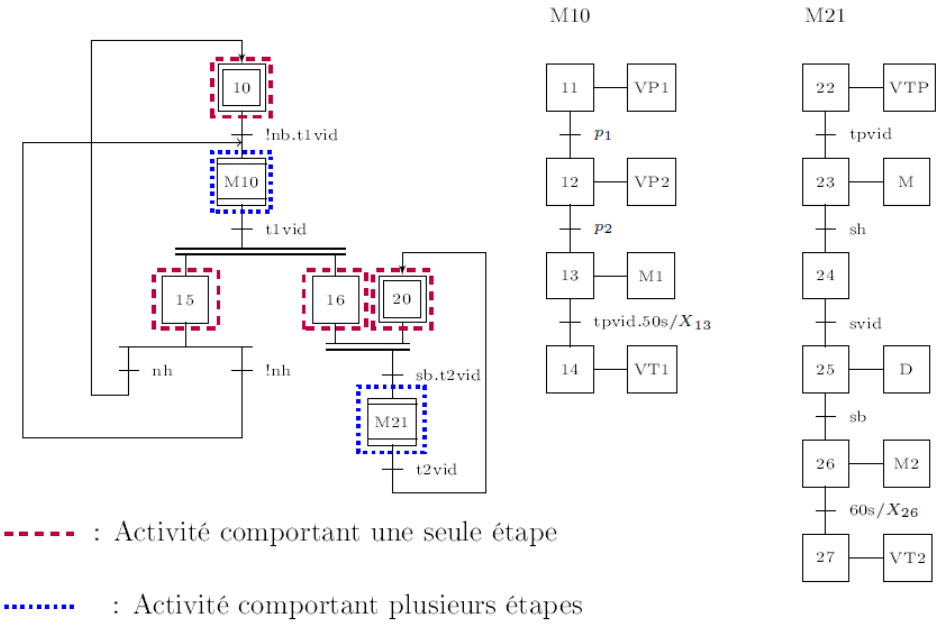


Figure 2.9 Découpage en activités du programme SFC de la Figure 2.8.

2.3.5.2 Optimisation des spécifications

Une *trace d'exécution* est une séquence d'évènements (changements d'état) correspondant à des changements d'état des capteurs et des actionneurs du système. Nous enregistrons ces occurrences par le biais du trafic réseau intercepté par les sondes. En raison de la périodicité de l'exécution des programmes des automates, notre observation n'est pas parfaite. Tous les évènements se produisant entre l'envoi de deux paquets successifs seront perçus comme simultanés à la date d'envoi du paquet. Appelons *trace observée* cette perception « réseau » de la réalité du processus. La Figure 2.10 illustre la différence entre la trace d'exécution et la trace observée (les évènements act↑ et act ↓ représentent le début, respectivement la fin de l'activité). L'intervalle temporel entre deux observations n'est pas régulier, il est un multiple de la période d'exécution du cycle de l'automate et de la période de scrutation des variables par le réseau.

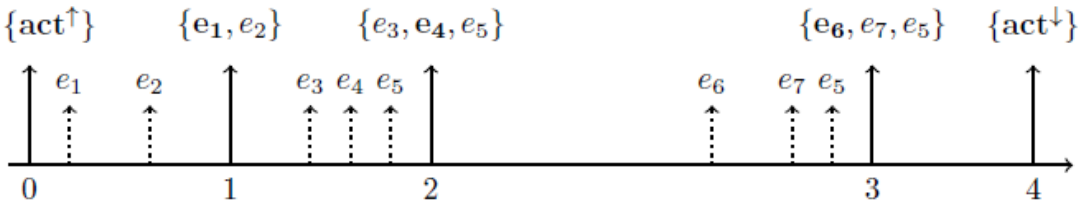


Figure 2.10 Evènements réels (flèches en pointillées) et évènements observés (flèches pleines)

En général, dans la fouille des spécifications, les patrons des spécifications sont utilisés afin de trouver des propriétés valides sur les traces [36], [37], [38]. Le principal problème rencontré est la grande quantité d'instanciations valides des spécifications. Ceci aura un impact négatif autant sur la complexité de la fouille que sur la détection. Nous avons conjecturé (et l'expérimentation l'a confirmé) que l'une des principales sources de prolifération des instances des spécifications est la superposition des portées. Considérons l'exemple de trace observée en Figure 2.11.

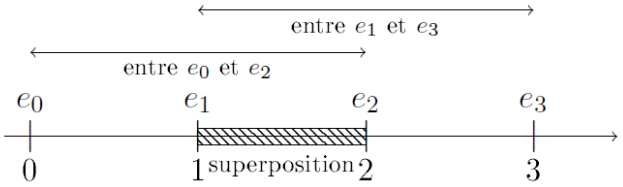


Figure 2.11 Exemple de superposition des portées

La fouille de la propriété de sécurité *absence de l'évènement A entre B et C* donnera lieu à plusieurs instances dont, par exemple, une entre 0 et 1, une seconde entre 1 et 2 et une troisième entre 1 et 3. L'occurrence de l'évènement A entre 1 et 2 donnera lieu à trois alertes basées sur le même patron de spécification. Avec un meilleur choix des portées (dans cet exemple on choisira (0,1), (1,2), (2,3)) ce phénomène sera évité. Nous avons proposé une procédure visant à optimiser le choix des portées pour la fouille des spécifications.

La procédure est présentée de manière informelle dans la section suivante. La description formelle a été publié en [35].

2.3.5.3 Fouille optimale

Définition 2-2 Une portée est une paire d'évènements (e_1, e_2) au niveau des capteurs/actionneurs

Définition 2-3 On appelle la couverture d'une portée (e_1, e_2) l'ensemble des intervalles (i, j) de la trace observée, tels que e_1 se produit à l'instant i et e_2 apparait à l'instant j.

Dans l'exemple de trace observée de la Figure 2.10, la couverture de la portée (e_1, e_6) est $\{(1,3)\}$ alors que celle de la portée (e_1, e_5) est $\{(1,2), (1,3)\}$.

L'objectif de notre procédure d'optimisation est de sélectionner un ensemble de portées S tel qu'il n'y ait pas de superposition entre les portées. Cet ensemble de portées doit respecter deux contraintes :

1) L'ensemble des portées couvrent collectivement la trace observée (« Pas d'angle mort »)
2) Toutes le propriétés valides seront fouillées (« Peigne fin »)

La première contrainte est assez évidente. Si des intervalles temporels de la trace ne sont pas couverts, aucune propriété de sécurité ne pourra être fouillée sur ces intervalles donc l'IDS sera aveugle entre ces instants. Dans la Figure 2.10 si l'ensemble des portées est $S = \{(act \uparrow, e\_4), (e\_6, act \downarrow)\}$, l'intervalle $(2,3)$ ne serra pas couvert et aucune propriété de sécurité ne serra fouillée et donc monitorée par la suite.

La propriété suivante garantit que l'intégralité de la trace sera fouillée sans superposition des portées.

Proposition 2-1. Un ensemble de portées S couvre de manière non-redondante l'intégralité de la trace observée si et seulement si chaque intervalle temporel $(i, i+1)$ de la trace en couvert par une seule portée de S

En effet, la *Proposition 2-1* garantit que l'intersection des couvertures des portées sera vide et que chaque intervalle entre deux observations successives sera couvert par une portée et seulement une.

La seconde contrainte (« Peigne fin ») est un peu moins intuitive. Supposons que, pour la trace observée de la Figure 2.10, on choisit $S = \{(act\uparrow, act\downarrow)\}$. Cet ensemble couvre l'intégralité de la trace et comme il ne contient qu'une seule portée, il n'y a pas de superposition de portées. Cependant il empêche la fouille de toute spécification autre que les spécifications universelles. Par exemple, si une propriété est valide uniquement sur l'intervalle (1,2), elle ne sera pas fouillée.

Intuitivement une propriété valide pourrait ne pas être fouillée si son intervalle de validité ne correspond pas exactement à un des éléments de la couverture d'une des portées. Nous introduisons la notion de « précision » d'un ensemble de portées qui caractérise la finesse du maillage des couvertures.

Proposition 2-2. Un ensemble de portées S est de précision maximale si et seulement si :

a) La couverture de chaque portée de l'ensemble contient un seul élément (un seul intervalle temporel couvert)
b) Pour toute portée (e_i, e_j) qui n'est pas en S et satisfait a) on peut trouver une portée en S dont l'intervalle temporel couvert est inclus dans l'intervalle couvert par (e_i, e_j) ou égal à lui.

Un ensemble idéal de portées (précision maximale, couverture totale sans redondances) qui satisfait à la fois *Proposition 2-1* et *Proposition 2-2* couvrirait donc chaque intervalle entre deux observations avec une portée unique. Un tel ensemble pour la trace observée de la Figure 2.10 serait $S = \{(act\uparrow, e_1), (e_2, e_3), (e_4, e_7), (e_6, act\downarrow)\}$. Remarquons que l'ensemble idéal peut ne pas être unique.

Afin de pouvoir générer automatiquement de manière efficace l'ensemble des portées, nous allons imposer une structure.

Définition 2-4. Un chemin dans une trace observée est une succession d'évènements observés tels que :

- A chaque instant temporel de la trace un seul évènement est associé
- Chaque évènement du chemin est associé à un seul évènement de la trace

A partir de la trace observée en Figure 2.10 un des chemins possibles est représenté en Figure 2.12. Le chemin choisi est $\{act\uparrow, e_1, e_4, e_6, act\downarrow\}$.

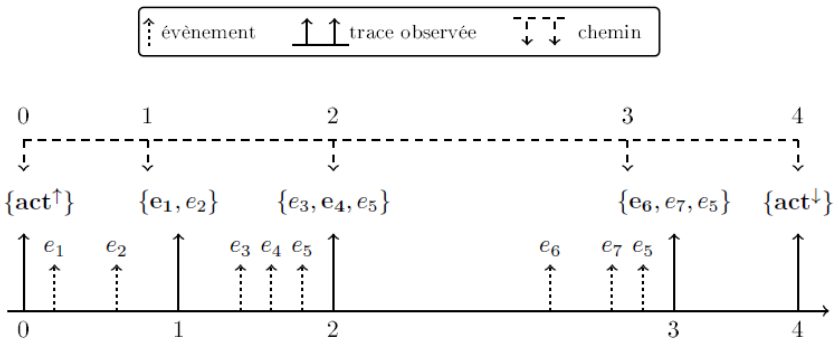


Figure 2.12 Exemple de chemin (non-unique) défini à partir d'une trace observée.

La propriété suivante relie la notion de chemin à celle d'ensemble idéal de portées.

Proposition 2-3 Pour une trace observée donnée, un ensemble idéal de portées satisfaisant *Proposition 2-1* et *Proposition 2-2* existe si un chemin existe.

L'Algorithme 2-1 permet de calculer l'ensemble des chemins à partir d'une trace observée t de manière incrémentale en construisant des chemins intermédiaires à partir des préfixes de t.

Algorithm 1: Procédure de génération des chemins

```
Data: t : Trace Observée
Result: P_t : Chemins générés à partir de t
P_t ← {};
for j ∈ t[0] do
    σ ← sequence();
    σ[0] ← j;
    P_t.add(σ);
for 1 ≤ i ≤ |p| − 1 do
    newPaths ← {};
    for j ∈ p[i] do
        for σ ∈ P_t do
            if j ∉ Range(σ) then
                σ[i] ← j;
                newPaths.add(σ);
    if |newPaths| == 0 then
        return None;
    P_t ← newPaths;
return P_t;
```

Algorithme 2-1 Construction de l'ensemble des chemins pour une trace observée donnée.

En général on utilise plusieurs traces d'exécution de la même activité pour la fouille des spécifications. La procédure de sélection des portées pour un ensemble de traces construit l'ensemble de chemins pour chaque trace observée et ensuite cherche un chemin commun à toutes les traces observées. L'ensemble des portées construit sur ce chemin est utilisé pour la fouille. La *Figure 1.1* illustre la procédure de sélection des portées pour un ensemble de traces.

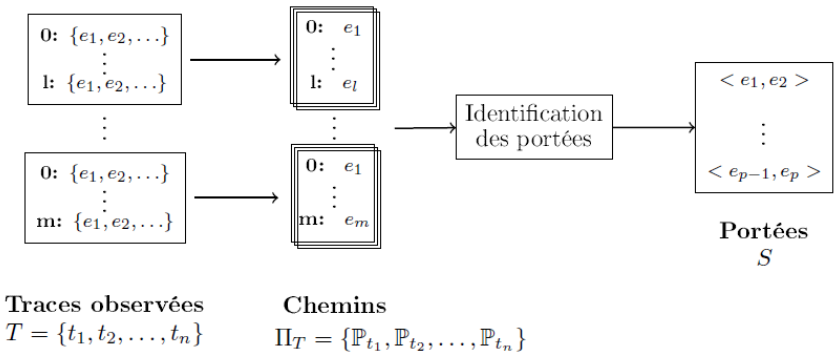


Figure 2.13 Choix des portées pour un ensemble de traces d'observation

L'analyse des conditions d'existence d'un chemin dans une trace d'exécution et des chemins communs à l'ensemble des traces sera discutée dans la section

2.3.5.4 Evaluation

Nous comparons notre approche avec l'une des références de la littérature [36]. Nous avons effectué de captures de trafic sans attaque sur un processus benchmark réalisé sur notre plate-forme expérimentale. Le processus et décrit dans l'annexe A.

Nous avons identifié 15 activités dans les processus de contrôle avec des durées entre 1 et 11 étapes SFC. La sélection des portées optimales a été faite sur quelques dizaines de traces d'exécution par activité (le nombre de traces n'est pas le même pour toutes les activités, il dépend de la fréquence d'exécution de l'activité).

Le résultat de la pré-sélection des portées en vue de la fouille des traces est présenté dans le Tableau 2-2. En général, les transitions entre les étapes du SFC sont conditionnées par un certain état des valeurs des capteurs ou des actionneurs. Cela veut dire qu'au moins un évènement de changement d'état des capteurs et des actionneurs se produit entre deux transitions. Donc, le nombre minimal de portées par activités doit être au moins égal au nombre d'étapes du SFC, ce qui est cohérent avec les chiffres du tableau.

Activité	# Etapes SFC	Temps (s)	# Portées
Act 1-1	4	0.064	6
Act 1-2	6	0.092	6
Act 2-1	4	0.084	6
Act 2-2	3	0.076	3
Act 2-3	3	0.069	3
Act 2-4	2	0.069	2
Act 3-1	11	1.508	17
Act 3-2	2	0.047	2
Act 3-3	1	0.060	1
Act 3-4	1	0.036	1
Act 3-5	1	0.040	1
Act 3-6	1	0.061	1
Act 3-7	1	0.035	1
Act 3-8	1	0.077	1
Act 3-9	1	0.063	1

Tableau 2-2 Résultat de la présélection des portées

Lors de la fouille des spécifications nous avons fouillé les patrons de sécurité spécifiés en [34]. Nous comparons avec la fouille sans pré-sélection des portées faite par Lemieux, Park et Beschastnikh [36]. Les résultats sont présentés dans le Tableau 2-3.

Activité	Propriétés qualitatives		
	Durée de la fouille (s)	Notre approche	LPB
Act 1-1	5	31	1616
Act 1-2	7	38	4247
Act 2-1	6	28	1437
Act 2-2	4.5	14	216
Act 2-3	4.5	14	409
Act 2-4	5	13	381
Act 3-1	32	281	46657
Act 3-2	5.5	12	105
Act 3-3	5	12	38
Act 3-4	6	12	38
Act 3-5	5.5	12	38
Act 3-6	6	12	38
Act 3-7	5.5	12	38
Act 3-8	6	12	38
Act 3-9	5	12	38

Tableau 2-3 Comparaison des résultats des fouilles entre notre approche et celle de [36]

Nous observons une réduction importante du nombre de spécifications identifiées, jusqu'à 150 fois pour l'activité la plus longue. La conséquence pour la détection est visible également : en phase de monitoring nous avons constaté une réduction de 10 à 30 fois du nombre de propriétés violées lors d'une attaque de séquence, pour les activités les plus longues. Cet impacte moindre sur la détection s'explique par le fait que les propriétés de sécurité ne sont pas toutes actives en même temps.

2.3.6 Détection

Dans la phase de détection nous déployons de moniteurs surveillant les propriétés de sécurités définies dans la phase des fouilles. Les moniteurs sont activés par un agent de reconnaissance des activités en ligne qui est le même que pour la phase de fouille des spécifications. Les violations de sécurité sont rapportées à l'opérateur, ainsi que la portée de la propriété de sécurité et l'activité courante. Le schéma global est présenté en Figure 2.14

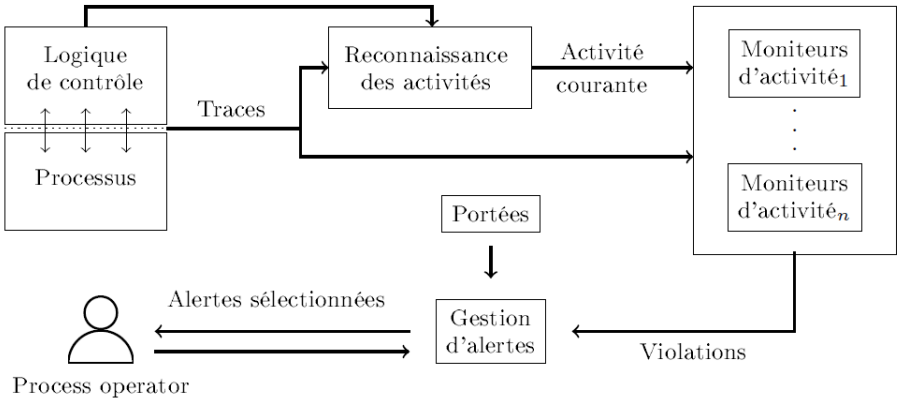


Figure 2.14 Schéma global du monitoring

2.3.6.1.1 Implémentation.

Nous utilisons une technique classique d'implémentation des moniteurs qui est celle des automates. Cette technique suppose la construction d'un automate non-déterministe alternant qui accepte un langage infini satisfaisant la formule LTL (automate de Büchi alternant [39]). Après quelques transformations l'automate est réduit à un automate fini acceptant les séquences qui violent la formule LTL [40]. L'autre technique est basée sur la réécriture des formule LTL après chaque évènement [41] [42]. La technique de réécriture utilise une technique de programmation dynamique pour garder l'historique de l'évaluation de la formule [43]. Généralement il est admis que le temps d'exécution est pénalisé par rapport à la technique basée sur les automates de Büchi [31] et c'est cela qui a guidé notre choix.

2.3.6.1.2 Evaluation

Nous avons déployé plusieurs attaques de séquence (5 types) consistant dans l'envoi de commandes qui violent les propriétés de sécurité. Sans surprise, toutes les attaques ont été détectées. Trois types d'attaques sur les cinq ont été détectées par un seul moniteur (donc pas d'alertes redondantes). Par comparaison les moniteurs dont les spécifications n'ont pas été présélectionnées lèvent 10 à 30 alertes par attaque pour les activités longues.

Le quatrième type d'attaque manipulait deux actionneurs afin d'accomplir l'action malveillante. Il a violé, en effet, deux propriétés de sécurité sur la même portée. C'est un phénomène attendu car à ce niveau nous n'avons pas encore réalisé de corrélation entre les alertes.

Enfin le dernier type d'attaque est le plus intéressant car il manipule un seul actionneur et il viole deux propriétés de sécurité sur des portées consécutives. En analysant les formules violées par rapport à la technologie du processus, nous avons constaté qu'il s'agissait en effet de la même propriété de sécurité

exprimée par une formule différente. Nous allons reprendre cette limitation de notre approche dans la section suivante.

Les détails des scénarios d'attaque et la liste des propriétés violées par chaque attaque sont présentés dans l'annexe B.

2.3.7 Synthèse des contributions, limites, recherche en cours et perspectives

En synthèse des sections précédentes, nous reprenons les deux contributions principales ayant fait l'objet des publications :

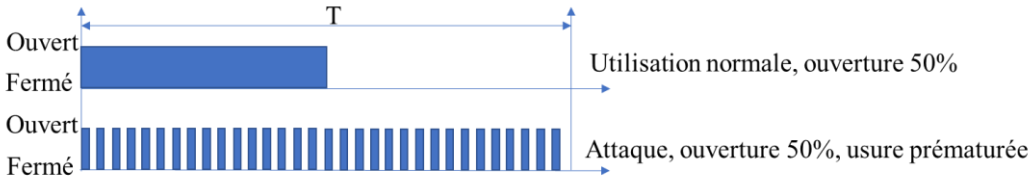
1) Le concept de détection des attaques orientées processus par couplage entre les domaines physique et cyber, en réalisant le lien entre les spécifications de sécurité du processus et le trafic réseau ainsi qu'une implémentation efficace par les techniques de vérification à l'exécution. Par rapport à d'autres approches de détection orientées systèmes industriels ([44] [45] [27] [46] [47] [48]) qui, soit se limitent uniquement au niveau des protocoles modélisant une certaine régularité des échanges, soit s'appuient uniquement sur des invariants globaux du système (en général valeurs maximales/minimales des variables) soit, enfin, nécessitent un coût d'implémentation élevé ([49]) nous pensons être le premiers à développer une approche qui prend explicitement en compte l'état du processus et sélectionne les propriétés de sécurité locales à surveiller. Ce résultat a été présenté en [50].
2) Le formalisme de présélection des portées pour la fouille des spécifications a un impact majeur sur la réduction du nombre de spécifications fouillées et surtout dans la réduction du nombre d'alertes en cas de violation des spécifications. Les expérimentations menées sont encourageantes, nous avons rencontré une seule situation qui engendre des alertes redondantes mais il serait intéressant d'élargir les expérimentations. Ce résultat a été présenté en [35].

2.3.8 Recherches en cours de publication

La thèse de Oualid Koucham est en cours de rédaction et une partie des résultats n'a pas encore été publiée. Nous présentons les principales contributions en cours de rédaction.

2.3.8.1.1 Détection des attaques quantitatives

Notre approche présentée précédemment est limitée à la détection des attaques quantitatives. Nous avons étendu l'approche aux attaques qualitatives. Ce sont des attaques qui violent des contraintes soit sur la spécification des durées de certaines actions (du type « mélanger deux produits pendant 30 minutes » ou « durée maximale d'utilisation du moteur inférieure à 10 minutes »), soit qui forcent l'usure prématurée d'un actionneur par enclenchements répétés sur une courte durée de temps (car pour certains actionneurs tout ou rien « l'ouverture partielle » se réalise en enclenchant l'actionneur pour une durée de temps proportionnelle à la commande par période). L'Exemple 2 montre une attaque d'usure sur l'ouverture d'une vanne. Dans le deux cas la vanne est ouverte à 50% (pas de violation de la spécification du processus) mais le second cas mène à une usure prématurée de la vanne.



Exemple 2 Utilisation normale d'un actionneur TOR (ouverture d'une vanne) versus une attaque d'usure. Dans le deux cas la vanne est ouverte à 50%

Cette approche utilise la même démarche générale que la détection des attaques qualitatives, mais elle s'appuie sur un formalisme diffèrent. Nous utilisons la logique temporelle MTL (Metric Temporal Logic [51], [52]). C'est une extension de la logique LTL qui rajoute des contraintes temporelles sur les

opérateurs. Ainsi, par exemple $G_{[1,5]}p$ signifie que l'expression p sera vraie entre le 1 et 5 unités de temps à partir de l'instant présent. MTL est également monitorable et des patrons de spécification existent. Trois catégories de patrons ont été spécifiés en [53] : imposant une durée maximale/minimale, une périodicité ou sur les temporisations des séquences temps réel (temps de réponse maximal ou durée minimale d'une action déclenchée). La Figure 2.15 présente les cinq patrons de spécifications quantitatives définis en [52].

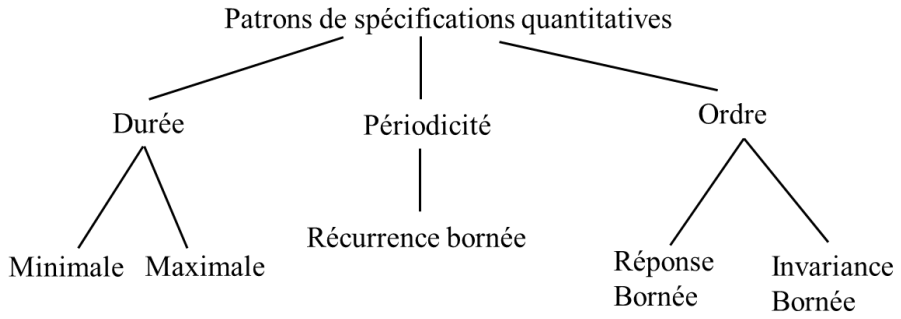


Figure 2.15 Patrons de spécification quantitatifs

Nous avons fouillé les patrons des spécifications de durées minimales et maximales qui sont significatifs pour les deux types d'attaques quantitatives. Nous avons mené des expérimentations sur le même processus avec des attaques quantitatives d'usure et sur les durées d'activités. Dans les deux cas les attaques ont été détectées de manière optimale (une seule alerte par attaque). Certes, c'est une première validation et des tests plus exhaustifs doivent être menés. Les

2.3.8.1.2 Corrélation des alertes multi domaine

Les contributions les plus récentes de la thèse d'Oualid Koucham se situent au niveau de la corrélation des alertes entre le domaine physique (moniteurs) et cyber (surveillance des flux réseaux, caractéristiques des échanges, télémétrie, etc).

Nous avons apporté deux contributions importantes :

1) Une technique de sélection des alertes pour la corrélation, permettant de réduire le nombre de fausses corrélations.
2) Une normalisation des alertes permettant la corrélation entre des détecteurs de natures différentes.

Trois motivations ont guidé cette étude. :

Réduction des faux positifs. L'une des limites de notre méthode de détection orientée processus est l'impossibilité de faire la différence entre les attaques et les interventions légitimes des opérateurs qui n'apparaissaient pas dans les traces d'exécution utilisées pour les fouilles.

En effet, la fouille des patrons de spécification peut aboutir à un ensemble de propriétés de sécurité plus restrictifs que les spécifications réelles. Si des actions rares n'apparaissent pas dans les traces, des propriétés « absence » résulteront de la fouille. C'est la principale source de faux positifs de notre approche.

Avec un nombre suffisamment élevé de traces d'exécution d'apprentissage nous pouvons espérer observer la quasi-totalité des comportements automatiques du système et réduire significativement cette source de faux positifs. Il est cependant très difficile d'observer tous les comportements possibles des opérateurs, car chaque opérateur aura ses habitudes et si, en mode manuel, l'ordre d'un certain nombre d'opérations est indifférent, le nombre des séquences de combinaisons d'actions légitimes ainsi que les valeurs des intervalles temporels entre les actions, sera très grands. Notons que, dans la littérature ([27],

[54], la modélisation typique des comportements des opérateurs se fait par de processus semi-Markoviens associant des distributions de probabilités aux chaines d'actions et aux durées.

La corrélation des alertes du domaine physique (moniteurs) et cyber permet de corroborer l'alerte physique avec, par exemple, une adresse réseau ou une interface inconnue ou encore avec une charge inhabituelle de paquets réseaux.

Surfaces d'attaque multiples et ambiguïté protocolaire. Les architectures de communication dans les systèmes de contrôle/commande modernes sont de plus en plus complexes. Très souvent les équipements (automates programmables, mais aussi les superviseurs) communiquent simultanément sur plusieurs interfaces situées sur plusieurs réseaux différents, éventuellement avec des protocoles IP et non-IP. Les protocoles industriels présentent une certaine ambigüité dans le sens que plusieurs requêtes différentes peuvent avoir le même effet sur une variable.

Pour illustrer ces propos, considérons un cas assez typique d'automate programmable communiquant par protocoles Modbus/TCP, Modbus/SOAP, EtherNet/IP, CAN. Un tel automate a quatre interfaces de communication car Modbus/TCP est considéré plutôt un protocole de niveau CIM 1 alors que EtherNet/IP et CAN se situent au niveau CIM0 et utilisent des supports physiques différents. Modbus/SOAP est un nouveau protocole utilisé plutôt pour accès à distance via un serveur web tiers et il est implémenté sur un module de communication séparé. Considérons une variable mémoire correspondant à la consigne d'un actionneur tout-ou-rien. En principe, cette variable peut être modifiée à travers n'importe quelle des quatre interfaces.

Plusieurs requêtes du protocole Modbus peuvent être utilisées pour modifier la même variable. Pour sa partie fonctionnelle, Modbus utilise un code numérique de fonction pour designer l'action et son objet (read/write one/multiple bit/register) une adresse, un nombre d'objets pour les actions sur plusieurs bits/registres et une valeur pour les actions d'écriture. Supposons que le bit mémoire considéré est %M41[11]. Sur certains automates (ABB par exemple) une adresse de type bit peut être vue comme un bit ou comme une partie d'un registre. Par exemple %M41 et %MW2.10 représentent la même variable : le 42ème bit de la mémoire[12]. Six codes de fonction peuvent être utilisés pour modifier la variable %M1 :

- Write single coil (fonction 5
- Write multiple coils (fonction 15)
- Write single holding register (fonction 6)
- Write multiple holding registers (fonction 16)
- Mask Write Register (fonction 22)
- Read/Write Multiple Registers (fonction 23)

En utilisant les fonctions d'écriture multiple (15, 16, 23) plusieurs requêtes ont pour effet la modification de la variable ciblée (par exemple l'écriture de 10 bits à partir de %M35 a aussi comme effet la modification du %M42). Il s'ensuit qu'une phase de prétraitement est nécessaire lors de la corrélation des alertes. Notons que l'utilisation de plusieurs protocoles et surfaces d'attaque a été utilisée par Industroyer/CrashOverride.

Reconstitution des scénarios d'attaque. Nous avons constaté lors des expérimentations (Section 2.3.5.4) que certaines attaquent peuvent violer les propriétés de sécurité de plusieurs moniteurs. Il est donc nécessaire d'utiliser la corrélation des alertes de plusieurs moniteurs afin de créer une unique méta-alerte.

[11] Nous utilisons la notation de la norme IEC 61131-3 : %M pour les variables mémoire de type bit, %MW pour les registres 16 bits.
[12] La numérotation des variables commence à 0, %M0 (ou %MW0.0) est le premier bit.

Approche. Nous nous appuyons sur les techniques classiques de corrélation des alertes [55], [56], [57]. Nous chercherons à corréler les alertes de nos moniteurs physiques avec les alertes du domaine cyber (inspection des flots, contenu de requêtes, propriétés de régularité du trafic).

La Figure 2.16 montre l'architecture globale de corrélation ainsi que le positionnement de nos contributions.

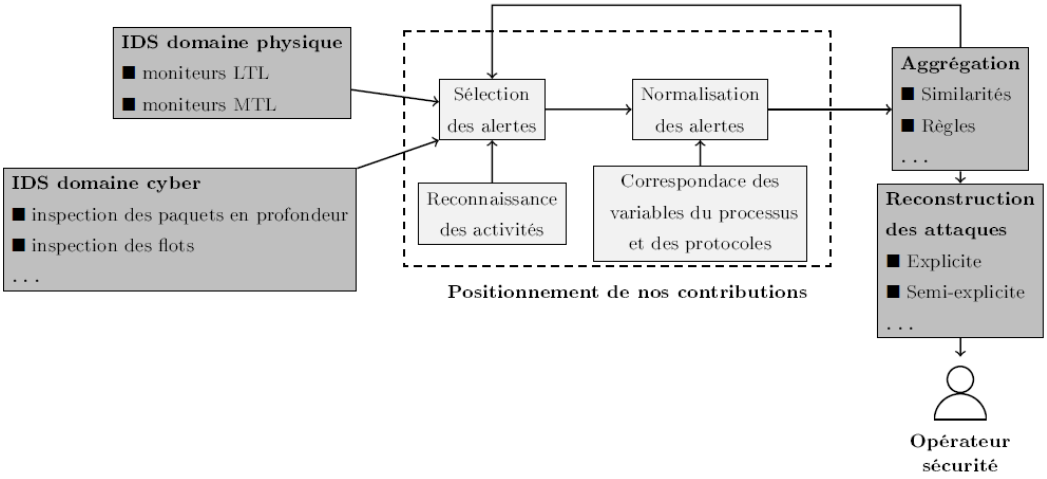


Figure 2.16 Vue globale de l'approche corrélation

Les éléments d'originalité se situent à deux niveaux :

1) Normalisation des alertes. Nous avons proposé un formalisme de normalisation des alertes afin de permettre la corrélation des alertes entre les domaines cyber et physique. Nous avons adopté une représentation des alertes basée sur un formalisme de premier ordre. L'objectif est de rendre les alertes du domaine physique compatibles avec les alertes du domaine cyber.
   La première étape de la normalisation est de mettre en correspondance chaque actionneur avec :
   - Les adresses réseau des automates qui peuvent les contrôler
   - Les protocoles qui peuvent être utilisés pour les modifier
   - L'identifiant de l'actionneur

   Par exemple, une alerte signalant l'activation de l'actionneur VTP1 (évènement VTP1↑) dont VTP1 est accessible par protocole Modbus à l'adresse 41 ou protocole DNP3 adresse 97 sur l'automate à l'adresse H_{PLC} sera translatée en $\uparrow \wedge \left(dstIp = H_{PLC} \wedge \left((protocol = Modbus \wedge address = 41) \vee (protocol = DNP3 \wedge address = 97)\right)\right)$.
   La deuxième étape associe les évènements sur les actionneurs avec des commandes des protocoles de communication. Par exemple pour l'activation de VTP1 à l'adresse 41 par protocole Modbus en utilisant les fonctions 5 ou 15 on obtient
   $(functioncode = 5 \wedge address = 42 \wedge data = 0xFF00) \vee (functioncode = 15 \wedge values[42 - startaddress] = 0xFF00 \wedge startaddress \leq c < startaddress + quantity)$
   La dernière étape consiste à marquer les attributs du domaine cyber qui ne sont pas significatifs pour le domaine physique (tels que le protocol_id et transaction_id dans le cas de Modbus/TCP).

   Suite à ces deux étapes, les alertes du domaine physique sont enrichies avec un sous-ensemble de champs permettant l'agrégation avec les alertes du domaine cyber.

2) Sélection des alertes. Nous réalisons une corrélation des alertes en ligne. La capacité et la durée de stockage des alertes étant limitées, nous mettons en place une technique de sélection des

alertes. Les approches classiques utilisent une fenêtre temporelle glissante de taille fixe. Cette technique n'est pas adéquate par rapport à notre approche orientée activité. En effet, une fenêtre de taille fixe risque soit de couvrir plusieurs activités et produire de fausses corrélations, soit stocker une trop courte mémoire des alertes et donc rater des corrélations valides. Nous avons proposé une sélection des alertes orientée activité avec trois politiques de sélection :

a. Sur toute l'activité. Nous utilisons la totalité des alertes envoyées depuis le début de l'activité pour les corrélations. Intuitivement on s'attend à ce que cette politique soit la plus performante en termes de nombre de corrélations correctes et nombre de corrélations ratées. En contrepartie, pour les activités longues le coût en terme de ressources de calcul risque d'être très élevé.

Nous avons donc cherché à maitriser le nombre d'alertes sélectionnées en utilisant deux autres politiques. D'abord, la mesure du temps en secondes n'étant pas adéquate pour notre cas, nous choisissons comme unité de temps l'intervalle (de durée variable) entre deux transitions dans le SFC. Les transitions étant conditionnées généralement sur les capteurs/actionneurs, cette mesure est observable. Les alertes sont donc groupées par étape du SFC. Nous définissons deux politiques de sélection partielle des alertes.

b. Etape adjacente. Cela revient à garder une mémoire d'une durée d'une transition dans le SFC. La source d'inspiration de cette politique est l'intuition du fait qu'un attaquant ciblera un état bien déterminé d'un processus, donc une certaine étape du SFC. On s'attend à ce que les actions malveillantes soient plutôt groupées.
c. Nombre d'étapes adaptatif. Lors de l'arrivée d'une nouvelle alerte, celle-ci est corrélée avec les autres alertes de l'étape courante. Si la corrélation est réussie, une corrélation est essayée avec les alertes de l'étape précédente. En cas de succès le processus continue en reculant dans le temps d'une étape à chaque fois jusqu'à ce qu'aucune corrélation ne puisse plus être trouvée.

Afin d'évaluer notre approche nous avons introduit deux métriques :

- taux de fausses corrélations (FCR) = #fausse_correlations/#nombre_de_correlations_produites
- taux de corrélations ratées (MCR) = #corrélations_ratées/#correlations_attendues

Les premiers résultats expérimentaux montrent que nos politiques sont équivalentes en termes de taux de fausses corrélations et nettement plus performantes que la fenêtre temporelle fixe (entre 0 et 2 % FCR, alors que la fenêtre fixe se situe entre 1% et 21 %). Pour les MCR la politique qui garde toutes les alertes de l'activité est la plus performante (entre 1% et 4% MCR) alors que les deux autres politiques se situent entre 14 et 34 % MCR. En général la métrique MCR est meilleure que celle de la fenêtre temporelle dont la performance est très dépendante de la durée de la fenêtre (nous avons fait varier cette durée entre 10s et 5 minutes). La différence significative de performance MCR entre la première politique et les deux autres prouve que les attaques peuvent avoir des manifestations intermittentes par rapport aux étapes du SFC, ce que mérite un approfondissement expérimental.

2.4 Perspectives de recherche

Les résultats obtenus dans cette thèse, autant dans la partie publiée que dans celle en cours de rédaction, confortent le paradigme central : celui de l'approche orientée processus pour la détection des intrusions. Les résultats expérimentaux, même s'ils ont été obtenus sur un système dont la partie processus est simulée, imposent comme une évidence le fait que l'interprétation du trafic réseau doit être mise en relation avec l'état du processus physique.

Je continuerai à développer des recherches dans la détection des intrusions orientées processus dans les systèmes industriels. L'approche par monitoring des propriétés de sécurité me semble tout à fait satisfaisante en termes de performance et je pense qu'elle restera l'outil principal d'implémentation. Plusieurs directions de recherche me paraissent particulièrement intéressantes :

1) Monitoring du comportement continu. Aller au-delà du comportement séquentiel semble le grand défi pour la suite directe de cette étude. La prise en compte du comportement continu nécessitera un nouveau formalisme qui devra tenir compte de la dynamique des variables continues. Plusieurs particularités du comportement continu rendent le défi scientifique intéressant :
    a. Le comportement continu d'un système industriel est piloté par un régulateur qui doit assurer la stabilité et l'asservissement d'une ou plusieurs variables continues. Les spécifications de sécurité du système peuvent être qualitatives (stabilité) ou quantitatives (temps de réponse, dépassement, erreur statique, bornes de la commande).
    b. En raison des contraintes temps réel fortes, les régulateurs sont le plus souvent embarqués sur l'équipement terminal lié aux capteurs/actionneurs par des entrées/sorties analogiques directes. La trace observée de l'exécution (construite à partir du trafic de supervision) fournira une image peu fidèle de la dynamique. En particulier, dans le cas de contrôleurs par retour d'état on n'aura pas accès à une partie des variables d'état.
    c. Les scénarios d'attaque sont plus compliqués car ils n'impliquent pas nécessairement la manipulation directe des actionneurs. Quelques possibilités alternatives sont : manipulation des paramètres des régulateurs ou manipulation des gains des cartes E/S (paramètres matériels).

A priori le formalisme candidat serait Signal Temporal Logic (STL) [58], [59], [60], qui étend la logique MTL avec des prédicats numériques sur des valeurs réelles. L'expressivité du formalisme permet d'exprimer des propriétés de sécurité basées sur les caractéristiques des systèmes dynamiques telles que positivité, dépassement maximum, stabilité, temps de réponse à 10%, etc. Dans [59] les auteurs présentent un exemple de propriété exprimant le caractère borné de la réponse d'un système, ainsi que sa stabilité et son temps de réponse (Figure 2.17).

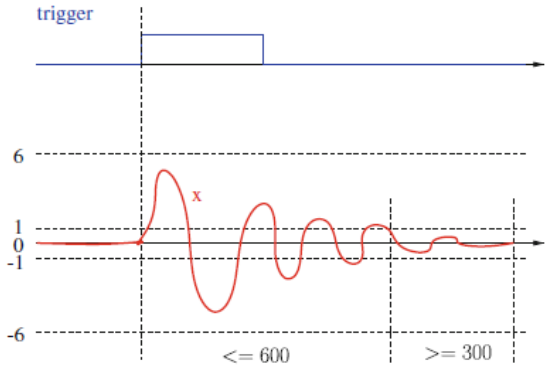


Figure 2.17 Propriété de stabilisation d'un signal

L'expression STL suivante exprime la borne maximale du signal à 6, un temps de réponse de stabilisation de 600 et la stabilité en absence d'entrées :

$$G\left(|x| < 6 \wedge \left(trigger \uparrow\rightarrow F_{[0,600]} G_{[0,300]}(|x| < 1)\right)\right)$$

A priori les propriétés de STL semblent adéquates pour notre champ d'application mais la définition des propriétés de sécurité des signaux continus asservis est à étudier ainsi que la monitorabilité, l'éventuelle définition des patrons de sécurité etc

2) Monitoring distribué. Dans notre étude nous avons supposé que nous puissions intercepter tout le trafic réseau nécessaire et nous avons défini des propriétés de sécurité globales du système.

Cependant, dans les installations réelles de grande taille, le réseau de communication est fortement segmenté afin d'optimiser le trafic.

Ces réseaux doivent être vus plutôt comme une multitude de réseaux locaux interconnectés dont les flux sont séparés logiquement. C'est une conséquence de l'application de la norme de cybersécurité industrielle [5].

Il serait intéressant d'étudier les propriétés de sécurité locales de chaque réseau et leur interdépendance afin de retrouver des propriétés globales. On réaliserait ainsi un monitoring distribué.

Un autre aspect important serait de déterminer le déploiement des sondes dans un tel réseau segmenté. Le nombre de segments peut atteindre facilement une vingtaine. Déployer une vingtaine de sondes sur un processus industriel peut atteindre rapidement un budget très conséquent. Il serait donc important d'étudier les dépendances entre les propriétés de sécurité locales en utilisant la connaissance métier du processus et essayer de minimiser le nombre de sondes et optimiser leur emplacement.

3) Corrélation des alertes avec les logs des équipements. Il s'agit de pousser encore plus loin la corrélation des messages entre les domaines physiques et cyber, en utilisant cette fois non seulement les alertes des différentes sondes, mais aussi les journaux d'évènements des équipements.

   Certains automates programmables modernes commencent à embarquer des capacités de journalisation des évènements (dont des évènements de cybersécurité) ou la capacité de les envoyer sur des serveurs syslog. D'une autre part, les logiciels SCADA journalisent les évènements systèmes et processus et ils stockent l'évolution des variables du système sur des serveurs d'historisation. Dans ce contexte il serait intéressant d'utiliser l'historique de processus dans la fouille des spécifications. Cela permettrait par exemple d'identifier clairement une partie des opérations manuelles réalisées par les opérateurs dans les traces d'exécutions exemptes d'attaques. Ensuite, la corrélation entre les alertes des IDS et les journaux des SCADA pourraient permettre d'identifier des attaques « d'aveuglement » des SCADA (dans le style de Stuxnet).

   Le défi consiste à développer un formalisme capable d'enrichir les journaux des équipements et SCADA avec les informations nécessaires permettant l'agrégation avec les alertes des IDS.

4) Langage de modélisation des systèmes industriels. Le formalisme que nous avons développé pour l'enrichissement des alertes physiques met en évidence le besoin d'un langage de modélisation des systèmes industriels permettant de décrire le matériel, le réseau de communication, les programmes embarqués, les capteurs/actionneurs et le lien entre les variables et les protocoles de communication. De langages répondant à une partie de ces besoins existent : OpenPLC propose une spécification pour les logiciels embarqués, SCL est un langage de modélisation des systèmes de distributions électrique, Cyber Physical Topology Language (CPTL) est un langage qui prend en compte simultanément l'architecture du système d'automatisme du smart-grid et celle du système de communication, mais qui n'a pas réussi à s'imposer. La spécification d'un DSML (Domaine Specific Modelling Language) permettant de surprendre les aspects cyber-physique avec prise en compte de la cybersécurité serait sans doute un outil très important autant pour des tâches de conception de l'architecture système que pour des activités plus techniques tels que la normalisation des messages et alertes de sécurité.

3 Cybersécurité des réseaux électriques

Ce chapitre décrit les résultats obtenus dans la thèse de Maëlle Kabir-Querrec (thèse CIFRE co-encadrée avec Jean-Marc Thiriet - Gipsa-lab) et le stage Master de Laurent Lê-Hebrard.

Les réseaux électriques constituent une catégorie à part parmi les systèmes industriels. Contrairement à l'automatisme industriel classique dont l'objectif est de piloter des processus industriels très divers aussi bien dans l'industrie alimentaire que dans l'industrie automobile ou chimique, l'automatisme des réseaux électriques s'adresse à un domaine dont les caractéristiques du processus physique sous-jacent sont relativement homogènes quel que soit le réseau et le pays. En effet, les composants, les technologies et les topologies des réseaux électriques sont relativement les mêmes dans tous les réseaux électriques.

3.1 Quelques notions d'électricité

Introduisons d'abord quelques notions de vocabulaire électrique pour fixer le champ d'application. La Figure 3.1 présente les principaux éléments d'un système de production et transport électrique.

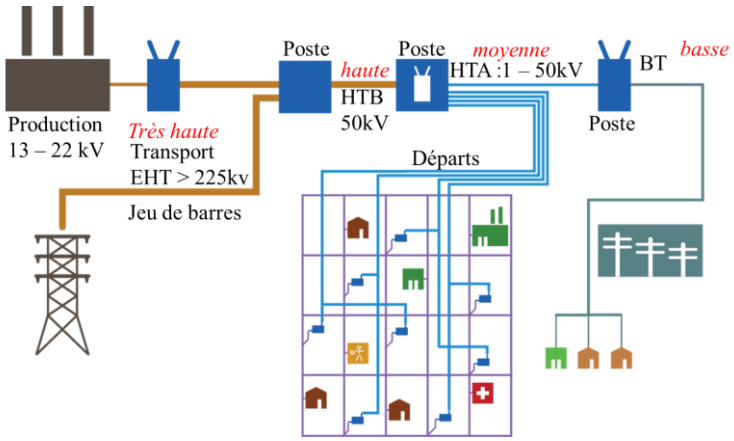


Figure 3.1 Transport et distribution électrique

Pour être efficace, le transport de l'électricité doit se faire à des tensions très élevés car les pertes en ligne sont proportionnelles au carré de l'intensité, donc transporter la même puissance a une tension plus élevée (et donc un courant plus faible) réduit les pertes.

Ainsi, la puissance produite par les centrales électriques est transformée à un niveau de tension de transport (Très Haute Tension ou Extra-Haute Tension)[13] supérieure à 225kV et est transportée sur de longues distances. A l'approche des consommateurs (villes ou zones industrielles), les niveaux de tension subissent des transformations vers des niveaux inférieurs de répartition (HTB) et distribution (HTA)[14] jusqu'au niveau de la basse tension (BT < 1000 V).

Toute la partie automatisme du processus est concentrée dans le poste électrique qui assure la transformation entre les différents niveaux de tension. Le poste électrique (Figure 3.2 reproduite depuis Wikipedia) est construit autour des transformateurs. Dans ce processus particulier, les actionneurs sont les disjoncteurs électriques dont le rôle est d'interrompre le courant électrique en cas d'incident. Des capteurs de courant et de tension permettent de mesurer l'état du réseau électrique et de détecter les éventuelles fautes. Bien évidemment, d'autres systèmes annexes sont déployés dans les postes. Les disjoncteurs HT sont eux-mêmes des procédés complexes comprenant une enceinte sous atmosphère isolante contrôlée pour interrompre rapidement les arcs électriques. Dans notre étude nous nous intéressons uniquement à la partie automatisme de poste c'est-à-dire aux fonctions de protection et contrôle du poste.

[13] Le terme normalisé est HTB bien que dans les usages EHT ou THT soient encore utilisés
[14] Historiquement on appelle Moyenne Tension la plage 1kV- 32kV

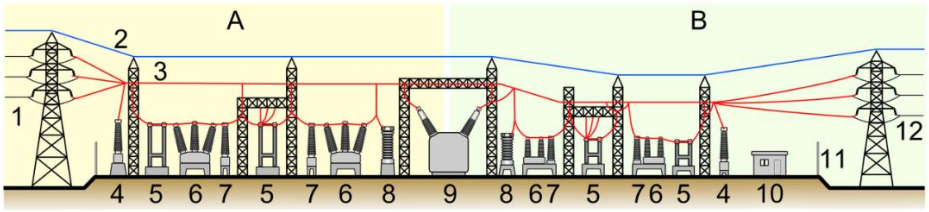


Figure 3.2 ***Composants électriques dans un poste*** *(A : côté primaire B : côté secondaire) 1. Ligne électrique primaire 2. Câble de garde 3. Ligne électrique 4. Transformateur de tension 5. Sectionneur 6. Disjoncteur 7. Transformateur de courant 8. Parafoudre 9. Transformateur (de puissance) 10. Bâtiment secondaire 11. Clôture 12. Ligne électrique secondaire)*

Dans les postes modernes, des calculateurs numériques embarqués implémentent des fonctions distribuées de protection, contrôle et mesure. Ces équipements sont appelés Intelligent Electronic Devices (IED). Les fonctionnalités à implémenter sont normalisées. La norme [61] dénombre 94 catégories de fonctions (avec un certain nombre de versions). Dans la pratique, les constructeurs chargent en usine le logiciel correspondant à un certain nombre de fonctionnalités (et prévoient les éventuelles cartes entrée/sortie nécessaires). L'utilisateur se limite à paramétrer les fonctions normalisées disponibles sur l'IED.

3.2 Système d'information des smart-grids

Le domaine de la distribution électrique a utilisé au départ les mêmes protocoles de communication que les autres systèmes industriels. Assez rapidement des modèles de données spécifiques ainsi que des protocoles dédiés ont été développés.

Le concept de smart-grid développée dans les dernières décades apporte, parmi d'autres, une intégration des systèmes physiques (la chaine production/transport/distribution/consommateur) des centres de contrôle (gestion de ressources distribuées et optimisation de la charge du réseau) mais aussi du modèle économique des marchés (échanges d'énergie, gestion de la demande et de la réponse) et des fournisseurs de services (gestion des consommateurs et de la consommation, tarification, etc). La Figure 3.3 [62] présente un modèle d'intégration réseau électrique/réseau de données et communication entre les différents domaines d'activités inclus dans les smart-grids.

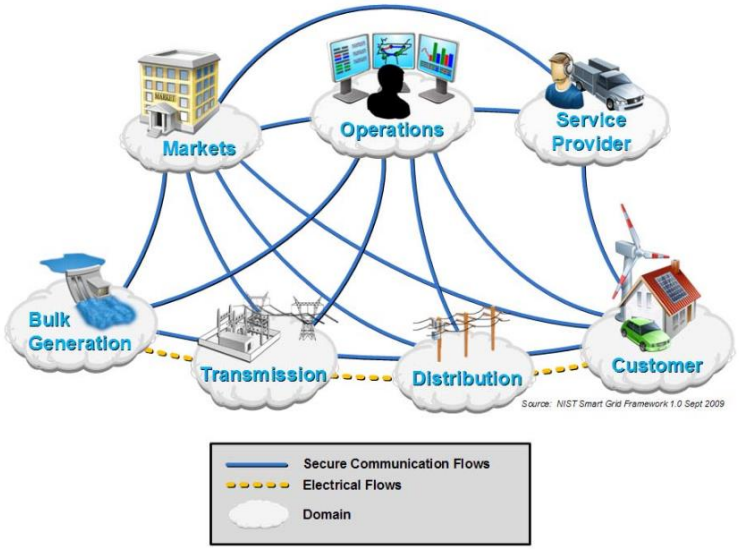


Figure 3.3 Représentation du modèle conceptuel smart-grid

La mise en œuvre d'un tel concept d'intégration sur une si grande échelle et intégrant autant d'acteurs et sous-systèmes a nécessité un énorme effort dans la direction de l'interopérabilité des échanges. Plusieurs normes ont été publiées au fil de temps. Un modèle général de données est fourni par le Common Information Model [63] et plusieurs autres normes connexes.

Parmi la multitude des flux d'informations nous nous intéressons aux flux de données proches du processus physique. Nous analysons donc les vulnérabilités et la détection d'intrusions dans les réseaux de communication des postes électriques de transformation. Ce choix est motivé par le fait que les attaques orientés processus vont cibler précisément ces réseaux. Ce fut le cas des attaques sur les réseaux Ukrainiens et notamment le cas de Industroyer/CrashOverrride.

3.3 La communication dans les postes CEI 61850

On considère généralement qu'il y a eu trois générations de protocoles de communication dans les postes électriques :

1) La génération « automation » avant la divergence dans le développent des systèmes de communication dans les réseaux électriques. Il s'agit des protocoles de communication industrielles classiques tels que Modbus ou, parfois, de protocoles propriétaires. Certainement quelques postes de ce type existent encore même s'ils sont censés avoir été remplacés avec les nouvelles générations.
2) La génération « SCADA ». Le nom prête à confusion, mais il s'agit des postes communiquant sur des protocoles dédiés aux réseaux électriques. Les protocoles sont spécifiés par la norme CEI 60870-5 [15]. La plus grande partie de postes existants communiquent en utilisant ces protocoles.
3) Génération « smart-grid » 61850. Les nouveaux postes sont supposés communiquer en utilisant les protocoles de la norme CEI 61850 [16].

3.3.1 La norme 61850

Les différentes parties de la norme ont commencé à être adoptées depuis le début des années 2000, ce qui en fait une norme encore en cours de déploiement et de développement par rapport aux durées de vie des équipements industriels. Ce ne sont pas uniquement les protocoles de communication qui sont spécifiés par cette norme mais aussi un modèle de données, un langage de modélisation (Substation Configuration Language – SCL) un formalisme pour les fonctions de protection et commande, des recommandations pour les réseaux de communication et même de spécification de compatibilité électromagnétique des équipements. Assez étrangement, l'aspect cybersécurité est totalement ignoré par la norme. Les aspects cybersécurité des smart-grids sont traités par une autre norme (CEI 62351) pour tous les protocoles de communication dans les smart-grids [64].

Dans le contexte de notre travail nous nous intéressons aux protocoles de communication spécifiés par la norme 61850 et le formalisme de spécification des fonctions de protection et commande.

3.3.2 Communication dans les postes

Un assez grand nombre de types d'IED existent selon les fonctionnalités standardisées qui sont embarquées. Certains IED sont spécialisés dans les mesures de courant, d'autres dans un certain type de protection électrique. La réalisation des schémas de protection électrique nécessite des échanges de données entre les IED. Trois types de trafic entre les IED sont identifiés dans les postes :

1) Le trafic des mesures de courant et tension. C'est un trafic avec de fortes contraintes temps réel. Actuellement, en Europe, l'échantillonnage des mesures de tension et courant se fait à 4KHz, donc toutes les 0,25 ms une trame contenant les mesures est envoyée. En raison de la criticité de ce trafic, un réseau dédié haute disponibilité (HSR ou PRP) [65] est préconisé. En raison des fortes contraintes temps réel, ce trafic n'est pas routé. La norme 61850 définit un protocole de communication appelé Sample Values (SV) sans acquittement (multicast) directement dans des trames EtherType. Habituellement un anneau PRP dédié (sans pontage) est utilisé.
2) Trafic des commandes temps réel. Certains schémas de protection sont compliqués et nécessitent des actions synchronisées entre plusieurs IEDs. L'exemple classique est la sélectivité logique de protections. Dans les réseaux électriques hiérarchiques (Figure 3.4), si un défaut se produit, le courant de défaut est observé par les plusieurs relais (deux dans la Figure

3.4). Afin de minimiser le nombre de consommateurs déconnectés, le relais aval disjoncte en premier et tente d'isoler le défaut. Il envoie au relais amont l'ordre d'attente. En cas d'échec il envoie un message au relais amont avec l'ordre d'ouverture du disjoncteur.

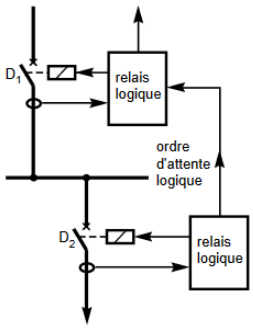


Figure 3.4 Schéma élémentaire de sélection logique

Ce type de trafic est aussi soumis à des contraintes temps-réel fortes. Le temps de propagation de bout-en-bout préconisé par la norme est de 3 ms. Il n'est pas routé non plus. La norme 61850 spécifie un protocole de communication (Generic Object Oriented Substation Event – GOOSE) qui est également une transmission multicast périodique EtherType temps réel. Un réseau HSR avec filtrage VLAN est préconisé, mais le trafic GOOSE peut cohabiter avec le trafic SV (l'encombrent réseau des GOOSE est bien plus faible).

4) Le trafic de supervision. La communication avec la salle de supervision est implémentée par un protocole porté par TCP/IP. Probablement pour des raisons de rétrocompatibilité, le protocole application choisi est MMS/TCP (Manufacturing Messaging Specification [66]). La norme 61850 a décidé d'adopter la version de MMS/TCP crée par Boeing en 1999 avec les couches hautes OSI transportées par TCP mais la version « complètement OSI » est permise (Figure 3.5).

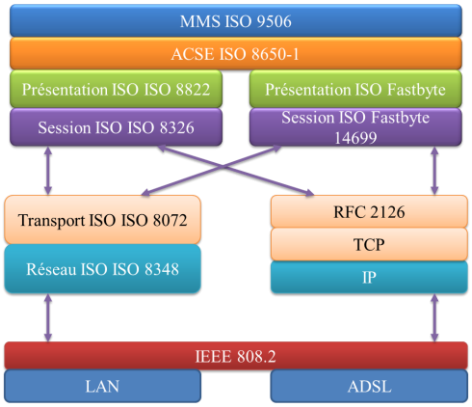


Figure 3.5 Combinaisons de couches permises pour le protocole supervision MMS.

Au niveau du poste on distingue ainsi la communication processus (réseau des SV), la communication horizontale (les commande GOOSE) et la communication verticale avec la supervision. Très souvent les interfaces de communication pour les différents flux sont physiquement séparées au niveau des IED. Généralement une double-interface PRP est dédiée aux SV, une interface simple ou double HSR est utilisée pour la supervision et les GOOSE. Eventuellement, une troisième interface est utilisée pour l'interface avec la station d'ingénierie. L'IED joue donc un rôle de pont entre les différents réseaux. La Figure 3.6 illustre les trois concepts de communication dans un poste électrique.

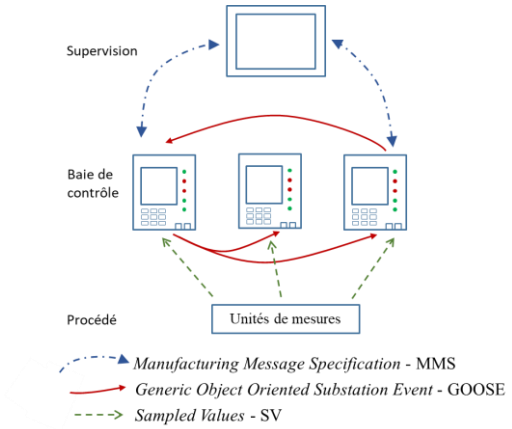


Figure 3.6 Les trois types de flux de communication au niveau de la baie.

L'esprit de la norme 61850 est d'implémenter les fonctions de protection et commande de manière distribuée. La brique de base s'appelle un nœud logique et c'est un objet (dans le sens de la programmation) qui implémente une fonctionnalité élémentaire (par exemple mesure de tension/courant, fonctions de protection, interface homme-machine). La spécification de la communication entre les nœuds logiques pour réaliser une fonction de protection utilise un formalisme appelle PICOM (Piece of Information COMmunication) qui permet de spécifier l'information à transmettre, sa source, sa destination, sa priorité et les contraintes temps-réel.

Dans les postes réels la gestion des flux de mesure est un vrai problème. A l'heure actuelle le standard de facto pour la communication dans les postes est FastEthernet ce que complique la gestion de la bande passante ; par exemple avec 18 points de mesure (18 Sample Values toutes les 0,25 ms) on sature le réseau. Il s'en suit que les réseaux de communications dans les postes sont, par ailleurs, très sensibles aux tempêtes Ethernet. Dans la Figure 3.7 nous montrons un exemple de distribution de flux de communication dans le cas de l'implémentation d'une sélectivité logique entre 5 protections. Pour une seule fonction distribuée on déploie 26 flots de communication dont 20 sont des multicasts.

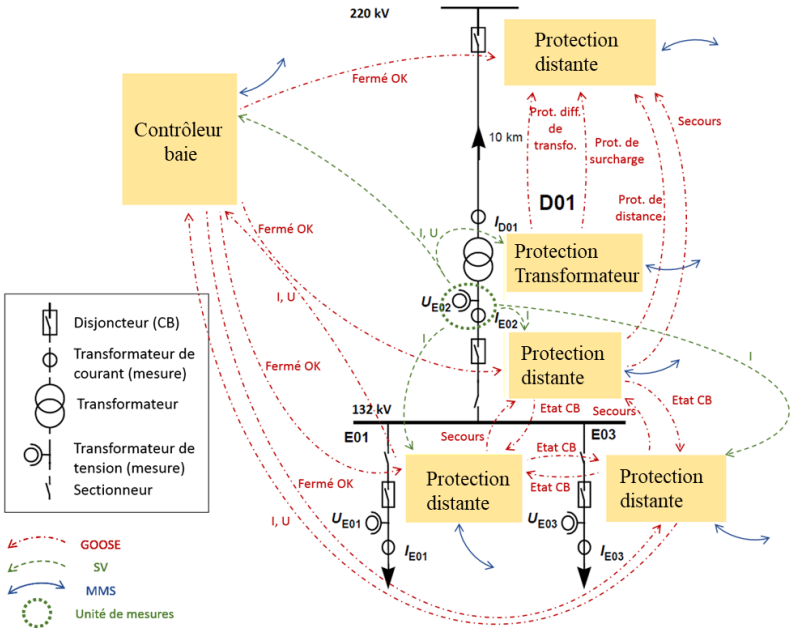


Figure 3.7 Exemple de distribution des flux dans un poste de petite taille

3.4 Nos contributions

Nous avons apporté trois contributions principales :

- Nous avons caractérisé une vulnérabilité décrite en [67] et écrit un algorithme de détection implémenté en Bro.
- Nous avons proposé une architecture résiliente face aux attaques sur les communications temps réel par fusion des alertes dans le SCADA.
- Nous avons spécifié un IDS générique dans le formalisme 61850

3.4.1.1 Attaque par injection de trames GOOSE.

Du fait de sa spécification, le protocole GOOSE présente une vulnérabilité facile à exploiter. Décrivons rapidement le fonctionnement du protocole avant de présenter notre mécanisme de détection.

D'un point de vue fonctionnel (en termes de réseaux électriques) GOOSE sert à signaler les changements d'état d'une ou plusieurs variables. La transmission étant de type multicast EtherType l'état de la/des variable(s) surveillée(s) est transmis périodiquement avec une périodicité réglable. En cas de changement d'état, le nouvel état est transmis tout de suite puis retransmis plusieurs fois après des intervalles temporels qui doublent à chaque fois jusqu'à la période normale. La Figure 3.8 illustre ce comportement (T0 est la période normale, T1, T2, T3 la période croissante après évènement).

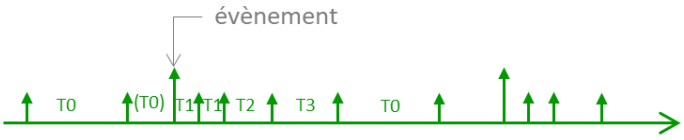


Figure 3.8 Fonctionnement normal des GOOSE

L'objectif de la rafale de GOOSE après un évènement est d'augmenter les chances que l'évènement soit signalé au plus vite au destinataire même si l'une des trames est perdue.

La transmission GOOSE n'est pas chiffrée ni authentifiée. Afin de permettre la distinction et l'ordonnancement des trames, deux compteurs existent dans la trame : State Number – StNum et Sequence Number – SqNum. A chaque changement d'état StNum est incrémenté et SqNum est remis à zéro. Entre deux changements d'état StNum est constant et SqNum est incrémenté avec chaque trame.

L'attaque consiste à insérer une fausse séquence de GOOSE, ce que revient à injecter un faux évènement. Notons qu'un mécanisme interne de protection des GOOSE contre les trames expirées fait que les trames dont la valeur StNum est inférieure à la Valeur StNum courante seront rejetés. Le système est donc incapable de récupérer l'état légitime après l'attaque.

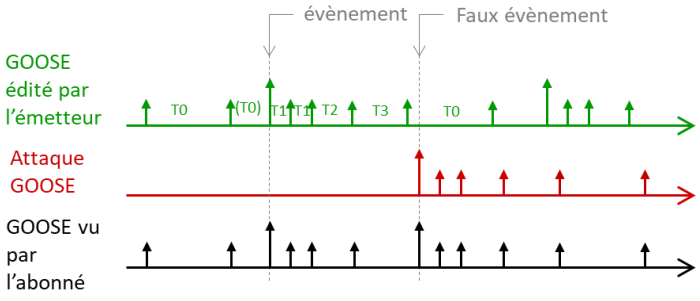


Figure 3.9 Illustration de l'attaque par insertion des faux évènements

Nous avons construit un banc de test et nous avons vérifié la faisabilité de l'attaque sur plusieurs IED du marché et plusieurs constructeurs. Nous avons constaté qu'aucun mécanisme de protection contre

ces attaques n'est implémenté. En pratique, une seule fausse trame suffit car la temporisation du mécanisme de « rafale » des trames en cas d'évènement n'est pas vérifiée par les IEDs.

3.4.1.2 Détection

Nous avons réalisé un détecteur qui vérifie pour chaque trame GOOSE la cohérence des tous les champs avec les spécifications et l'application, ainsi que la cohérence temporelle de séquences et la cohérence des évènements. Chaque évènement surveillé (chaque flux de trames GOOSE) est décrit dans le fichier de configuration de l'émetteur écrit en formalisme SCL (Substation Configuration Language). Nous extrayons ces informations à partir des fichiers de configuration des IED afin de paramétrer l'IDS.

1) Pour chaque trame l'IDS vérifie :
   - Les valeurs des champs de protocole (utilisation des VLAN, octets réservés, adresses multicasts, longueur de la trame, nombre d'évènements dans le jeu des données)
   - Les flux : correspondance des adresses MAC émetteur et destinataire avec l'application.
2) Pour chaque paire de trames on vérifie la cohérence des numéros de séquence et d'état, ainsi que la valeur de l'intervalle entre deux trames se situe entre les limites prédéfinies (Algorithme 3-1)

Algorithme 3-1 : Vérification du séquencement des GOOSE et des intervalles entre deux trames

if $StNum_n == StNum_{n-1}$ **then**
  if $SqNumn \neq SqNum_{n-1} + 1$ **then**
    Alert(CountAlm), Log
  else if $(SqNum_n == \{1,2\} \wedge \Delta T_n \neq Filter(GoID).MinTime) \vee (SqNum_n \geq 3 \wedge [\Delta T_n \neq Filter(GoID).MaxTime \wedge \Delta T_n \neq 2 * \Delta T_{n-1}] \vee [\Delta T_n \neq Filter(GoID).MaxTime])$ **then**
    Alert(TxAlm), Log
  end if
else if $StNum_n == StNum_{n-1} + 1$ **then**
  if $SqNum_n \neq 0$ **then**
    Alert(CountAlm), Log
  else if $\Delta T_n \geq MaxTime$ **then**
    Alert(TxAlm), Log
  end if
end if

3) Détection des attaques par faux GOOSE. Vu le type d'attaque, si les trames forgées sont parfaites (*i.e.* elles passent les deux vérification précédentes), la seule anomalie visible au niveau cyber apparaitra lors de l'arrivée de la première trame légale après l'attaque. Afin de détecter l'attaque de manière idéale il faut :
   a. Garder dans l'historique les derniers couples (StNum,SqNum,valeur_état) observés pour chaque variable surveillé.
   b. A chaque trame arrivée si $StNum_n < StNum_{n-1}$ vérifier si le champ de SqNum de la trame arrivée se situe dans la continuité de la séquence mémorisée et si oui, lever une alerte.

   Risque de faux positifs : si la dernière trame avant un changement d'état est retardée et l'ordre est inversé, c'est-à-dire la trame avec les compteurs (Stnum,0) arrive avant la trame (Stnum-1,Sqnum) l'IDS va lever une fausse alerte. Cette situation, bien que possible, est assez peu probable si le réseau est correctement dimensionné.

   On peut imaginer aussi que l'attaquant au lieu de forger un faux évènement va forger la trame qui déclenche l'alerte du IDS. Dans les deux cas (faux évènement ou faux « non-évènement »)

le système est sous attaque, donc ce n'est pas vraiment un faux positif. Il faut cependant en tenir compte dans l'interprétation des alertes.

Implémentation et évaluation. Nous avons fait le choix d'intégrer cet algorithme de détection à l'IDS Bro en tant que module. Nous avons testé la détection dans les conditions extrêmes de génération des GOOSE (2000 trames/seconde donc l'équivalent 24 Mbit/sec par application GOOSE). Nous avons souhaité un temps de décision de l'IDS inférieur à 3ms (contrainte sur le temps de propagation de bout-à-bout des GOOSE). Pour les réseaux habituels dans les systèmes industriels (FastEthernet) cela revient à traiter 4 flux GOOSE de 2000 trames/seconde en 3ms. Il s'est avéré que Bro sature rapidement la mémoire face à ces débits et la quantité d'informations à utiliser. La limite actuelle de l'implémentation est le traitement d'environ 50 Mbit/s en moins de 3 ms donc 2 flux GOOSE à 2000 trames/sec ou 4 flux à 1000 trames/sec. Nous pouvons donc traiter la moitié de la capacité maximale d'acheminement du réseau. Ces travaux ont été publiés en [68], [69].

A l'avenir il serait intéressant de tester l'algorithme indépendamment du Bro. Vu la simplicité des algorithmes de détection (même s'il y a une recherche dans l'historique de flux) une implémentation dédiée pourrait améliorer la capacité de traitement.

Nous avons fixé le temps cible de détection à moins de 3ms afin de nous permettre de déployer une contre mesure de l'attaque le plus rapidement possible. Nous avons proposé, par la suite, une architecture résiliente aux attaques basée sur le concept de l'intégration du SCADA et du SIEM.

3.4.1.3 Architecture résiliente

Dans l'esprit du même paradigme de la modélisation cyber-physique, une partie de travaux de la thèse de Maëlle Kabir-Querrec ont investigué la convergence entre les informations du processus (SCADA) et les informations de cybersécurité (SIEM). L'idée sous-jacente est que, en cas d'attaque cyber, celui qui doit prendre la décision opérationnelle (arrêt du processus ou mise en sécurité, isolation ou pas d'une partie du système attaqué, changement du mode opératoire) est l'opérateur de la salle de contrôle du processus physique. Seule la personne ayant la connaissance métier du processus physique peut prendre la responsabilité des actions à exécuter en cas de disfonctionnement (dont cyber-attaque).

L'approche que nous avons proposée en [70] est basée sur trois concepts :

1) La « double programmation » des IED : un mode de repli en absence de communication est prévu sur les IED
2) Le « double réseau » : un réseau séparé est utilisé pour la remontée des alertes de sécurité et pour la supervision.
3) La convergence SIEM/SCADA : les alertes de sécurité sont remontées au SCADA. En cas d'attaque, une règle automatique déclenche l'envoi par le SCADA de l'ordre de passage en mode repli des IED

3.4.1.3.1 Architecture exemple

Nous avons appliqué notre démarche sur un schéma de protection classique de couplage de deux jeux de barres (Figure 3.10). Dans ce schéma chaque section (jeu de barres) alimente plusieurs départs. En absence de défaut, le couplage est ouvert. En cas de défaut sur un des jeux de barres, le défaut est isolé et la commutation automatique (fermeture du couplage) permet au second jeu de barres d'alimenter les départs isolés. L'ordre de fermeture du couplage est envoyé par l'IED ayant isolé la faute (IED1 ou IED2 en Figure 3.10) via une communication GOOSE.

La communication GOOSE pouvant être attaquée de deux manières (par tempête Ethernet et par injection de fausses GOOSE), nous avons déployé deux sondes : une basée sur les outils habituels de

mesure de trafic (ifstat en occurrence) la deuxième basée sur notre détection de trames GOOSE corrompues. Le SCADA surveille périodiquement l'utilisation du réseau GOOSE et reçoit les alertes de l'IDS.

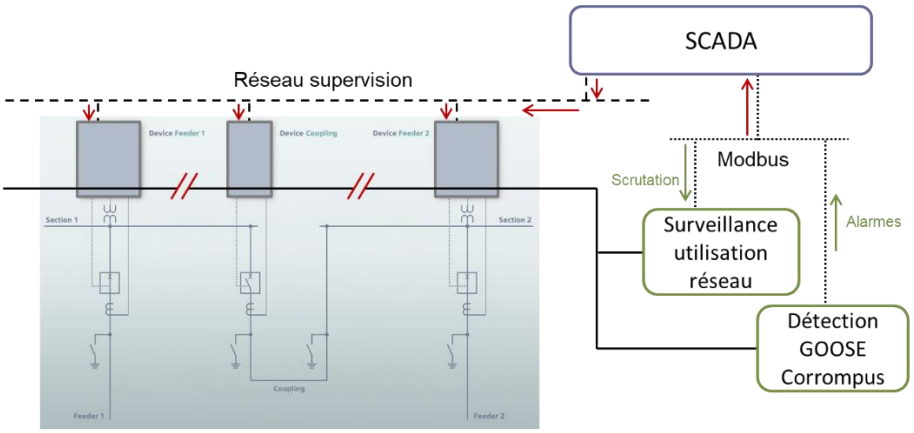


Figure 3.10 La maquette de test de l'architecture résiliente

En cas de dépassement de l'utilisation maximale prédéfinie du réseau ou si le détecteur remonte une alerte, le SCADA envoie une commande aux IED via le réseau de supervision forçant le passage en mode repli.

Programmation des IED. Le mode repli pour ce schéma de couplage de jeux de barres est particulièrement simple. En cas de doute sur la légitimité des commandes il suffit d'inhiber la fermeture du disjoncteur de couplage. La seule difficulté de déploiement réside dans les temps de réaction.

Le temps de réponse général d'un IED 61850 est de 4 ms selon la norme (3ms délai de transmission, 1 ms temps de réponse application). Bien qu'on impose à notre IDE un temps de détection inférieur à 3ms, le SCADA ne peut pas réagir dans 1 ms. Cependant, le temps de réponse des systèmes de protection est bien moins contraint. En pratique les délais acceptables sont de l'ordre de 30 – 40 ms exceptionnellement jusqu'à 1s [71]. Nous insérons un délai de réaction de 10ms dans le programme de l'IED entre la réception du GOOSE et la fermeture du disjoncteur.

La généralisation de l'approche nécessite une analyse au cas par cas des schémas de protection usuels. Cette étude est évoquée dans la partie Perspectives de ce chapitre.

3.4.1.4 Extension cybersécurité de la norme CEI 61850

La norme 61850 ne spécifie ni d'objet (nœud logique), ni de fonctionnalité orienté cybersécurité. Dans la même logique de convergence des informations processus et cybersécurité, nous avons spécifié des nœuds logiques orientées détection des intrusions. L'objectif est double : d'une part il s'agit d'intégrer les sondes dans l'environnement CEI 61850 et ne pas multiplier artificiellement les protocoles (dans l'exemple d'architecture résiliente de la section 3.4.1.3 nous avons utilisé Modbus pour la communication avec le SCADA à défaut des nœuds logiques 61850 spécialisés). D'une autre part si des composants de cybersécurité sont à déployer sur un IED, cela évite le déploiement de plusieurs piles de protocoles dans le système.

La fonction IDS que nous avons définie utilise sept nouveau nœuds logiques (les nœuds CY* en Figure 3.11) que nous avons introduit et qui correspondent aux fonctionnalités élémentaires de l'IDS : interception des trames, analyse des trames, vérification des champs de protocole par trame ou par séquence, vérification des valeurs des champs du protocole par trame et par séquence et, enfin, la remontée des alarmes. L'affichage des alarmes utilise des nœuds 61850 standard permettant ainsi l'intégration transparente dans un SCADA. Nous avons défini également les spécifications de flux de données entre les nœuds logiques (PICOM). L'approché a été publiée en [72]. La spécification complète est présentée dans les annexes de la thèse de Maëlle Kabir-Querrec [73].

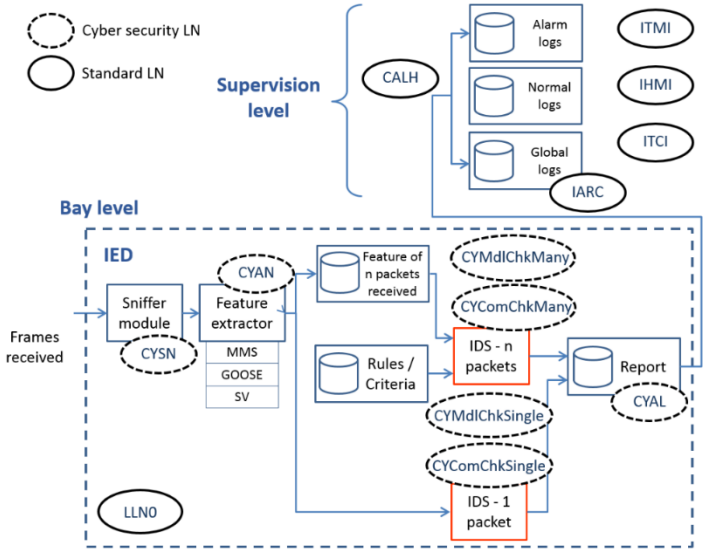


Figure 3.11 Schéma global de la fonction IDS

Pour l'instant nous n'avons pas encore étudié l'implémentation de la fonction. L'implémentation aurait dû faire l'objet d'un projet avec le partenaire industriel de la thèse, projet dont le financement n'a pas abouti. Néanmoins, nous pouvons mettre en évidence les avantages attendus de l'approche et les potentielles difficultés d'implémentation.

Avantages :

- Chaque IED peut jouer le rôle d'une sonde, détecter des attaques et déployer des contremesures. C'est particulièrement avantageux dans des applications de type architecture résiliente (Section 3.4.1.3) où, dans l'exemple de la Figure 3.10, l'IED « couplage » détecte directement l'attaque sur le protocole GOOSE et commute son fonctionnement en mode repli. On s'attend à un gain de capacité de traitement des flux GOOSE. Chaque IED pourrait analyser uniquement les flux qui lui sont destinés.
- L'approche 61850 permet de distribuer ou pas les fonctions. Les spécifications PICOM peuvent se traduire par un flux réseaux ou par une communication entre deux nœuds situés sur le même IED physique. La fonction IDS pourrait être distribuée sur plusieurs IED.
- Notons que le formalisme n'est pas dédié uniquement aux communications GOOSE. Il s'applique à tout type de communication 61850 (GOOSE, SV ou MMS).

Potentielles difficultés :

- La difficulté principale est, clairement, la consommation des ressources de calcul des IED. Pendant longtemps il était inimaginable de déployer des composantes logicielles autres que celles de contrôle/commande de processus. Les processeurs des dernières générations des IED sont basés sur des FPGA, ce qui permet, à priori, le déploiement des processeurs optimisés pour l'application. Certains constructeurs ont commencé à déployer des composantes de cybersécurité et de remontée des alarmes (principalement pour le contrôle d'accès par l'interface programmation et vérification de la signature des logiciels embarqués). Il devient donc envisageable de déployer des composantes de sécurité plus poussées.
- La distribution de la fonction IDS sur plusieurs IED nécessite une analyse préalable des volumes de données à échanger. Par exemple si l'IDS est concerné par la détection des trames GOOSE corrompues et si le trafic est élevé, il ne serait pas réaliste de distribuer les nœuds interception, analyse et vérification des trames et séquences sur des IED différents.
- La remontée des alertes doit se faire sur une interface dédiée. Pour l'instant nous supposons que les alertes sont remontées par l'interface de supervision. Une solution alternative serait d'utiliser

une interface supplémentaire pour la transmission des alertes vers un nœud de génération des alertes (CYAL) qui communiquerait par un canal sécurisé avec le SCADA.

La perspective de ce volet de travail est la réalisation d'un prototype intégré à un environnement 61850 et l'évaluation des performances.

3.5 Perspectives de recherche

Dans mes recherches futures dans le domaine de la communication dans les réseaux électriques, je poursuivrai l'idée de défense du processus physique, plus précisément des fonctions de protection et commande. Deux directions de recherche principales se situent en prolongation directe des résultats précédents.

Modélisation et analyse des risques des fonctions de protection. RTE vient de rendre publique la spécification 61850 [74] des fonctions de protection, commande, exploitation, surveillance et des fonctions générales de tranche, ainsi que les modèles d'équipements et d'interfaces qui seront utilisés dans tout nouveau poste du réseau de transport français. Ce document fixe les limites de l'étude et permet l'analyse des risques et la génération des scénarios d'attaque possibles. Considérons par exemple une fonction de protection (Protection Antenne Passive – LDPAP). La spécification RTE indique la nature des flux de données utilisés et leur fonctionnalité (Figure 3.12)

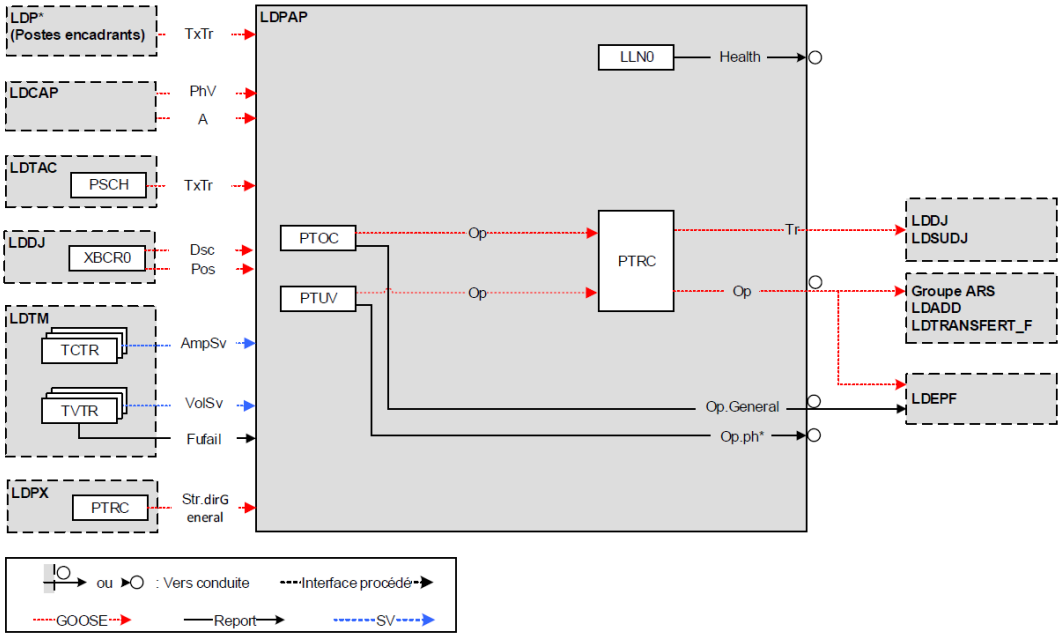


Figure 3.12 Spécification des flux de données et des nœuds logiques d'une fonction de protection.

L'analyse des risques et l'identification des attaques possibles est quasi immédiate en utilisant une analyse de type arbre de causes des évènements. Prenons l'exemple de l'évènement indésirable « déclenchement intempestif d'un disjoncteur (signal GOOSE Tr[15]) » et déroulons une analyse simplifiée. Une première possibilité d'attaque est évidemment l'insertion d'un faux évènement (trames corrompues) dans le flux GOOSE Tr. En déroulant en arrière le processus de création du signal Tr on observe qu'il est généré par le nœud logique standard PTRC[16] qui se déclenche sur activation d'un des flux GOOSE Op. Une deuxième possibilité d'attaque est donc l'insertion d'un faux évènement GOOSE dans l'un des flux Op. En suivant l'arborescence des signaux, les flux Op sont générés par les nœuds

[15] Tr (Trip) est l'ordre de déclenchement d'un disjoncteur
[16] Protection trip conditioning

PTOC respectivement PTUV[17]. Ces fonctions de protection utilisent des mesures de courant, respectivement de tension, pour décider du dépassement du maximum de courant, respectivement de tension. Plusieurs sources des mesures sont possibles selon l'implémentation physique concrète (mesure directe, Sample Values, ou transmissions d'un autre poste par GOOSE). Supposons que dans l'implémentation réelle les Sample Values sont utilisées. La troisième possibilité d'attaque est d'injecter des fausses valeurs SV dans le flot des trames Sample Values. Les trames Sample Values sont assez similaires aux GOOSE, car elles utilisent aussi des compteurs de trames, donc des attaques par corruption des valeurs des compteurs sont possibles. Ces différents scénarios sont représentés dans l'arbre de causes de la Figure 3.13.

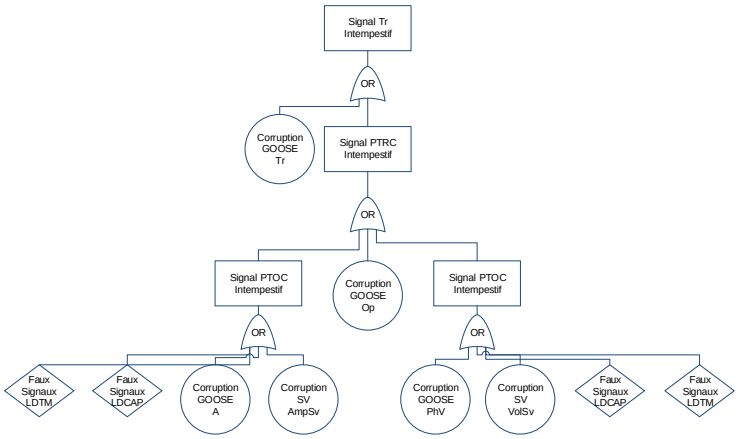


Figure 3.13 Analyse par arbre de causes des scénarios d'attaque sur la fonction de protection LDPAP

L'objectif est d'automatiser l'analyse des fonctions de protection spécifiées ainsi que la génération des scénarios d'attaque simples (tel que dans l'exemple précédent), d'aborder l'étude des architectures complexes combinant plusieurs fonctions de protection et ensuite de générer des attaques complexe par combinaison des attaques basiques.

Analyse des vulnérabilités des protocoles 61850. Dans la même démarche que pour le protocole GOOSE, j'ai l'intention de développer des recherches sur les vulnérabilités des protocoles SV et 61850. Le protocole SV présente le même type de vulnérabilités que GOOSE (corruption des compteurs) mais, en plus, une attaque sur la temporisation des trames devrait être possible. En effet, chaque trame porte un échantillon du signal mesuré, mais les trames ne sont pas horodatées. Une attaque sur la source de temps des unités de mesure de courant et tension serait à priori envisageable. Sur le protocole MMS quelques tests ont montré que des corruptions des champs des protocoles intermédiaires OSI (en particulier ACSE 8650-1) sont possibles.

Langage de modélisation des réseaux électrique. Dans la même logique que pour les systèmes industriels génériques (Section 2.4 - 4), j'ai l'intention de développer un langage de modélisation généralisant le langage proposé par la norme (SCL – Substation Configuration Langage) permettant d'inclure, en plus de la description matérielle et logicielle des IED, la configuration des fonctions de protection distribuées, les flots de données et les signaux échangés. L'idée est de fournir des outils d'automatisation de la création des scénarios d'attaque pour des architectures complexes.

[17] Protection Time OverCurrent respectivement Protection Time UnderVoltage, détection temporisée de dépassement du maximum de courant, respectivement minimum de tension.

4 Recherches connexes : plateforme expérimentale et performances des réseaux de communication industriels

Depuis 2014, je développe une plateforme expérimentale[18] d'interopérabilité et cybersécurité des réseaux de communication industriels couvrant les domaines de l'automatisme industriel, de la distribution électrique, ainsi que du tertiaire. La plateforme est hébergée par l'ENSE3[19], école de Grenoble-INP, étant également utilisée dans la formation des élevés ingénieurs dans les domaines de la communication industrielle, du temps-réel, des bases de données pour l'électricité et l'automatisme. Le matériel disponible réunit une centaine d'automates programmables, relais de protection, superviseurs industriels, capteurs communicants, interfaces homme-machine industrielles couvrant la plus grande partie des protocoles de communication du marché et les constructeurs les plus importants en France.

Dès le départ, la plateforme a été conçue en vue d'une utilisation flexible, reconfigurable, sans être liée à un processus physique ou un domaine applicatif particulier. Un point crucial est la possibilité de déployer des attaques sur les infrastructures de contrôle-commande sans courir le risque d'endommager un processus physique. J'ai donc développé une approche hardware-in-the-loop permettant d'interfacer les vrais équipements industriels de contrôle-commande avec une simulation de processus. Une carte électronique microcontrôleur est utilisée pour simuler les capteurs et les actionneurs. La carte est liée via des cartes d'adaptation du signal aux entrées/sorties des automates programmables et communique avec une simulation logicielle. Afin de permettre un maximum de flexibilité et le passage à l'échelle, la communication entre la carte électronique et la simulation se fait via un protocole UDP/IP simple. La carte peut ainsi s'interfacer avec n'importe quel simulateur cadencé (Matlab/Simulink Temps-Réel, Modelica) ou avec une interface de pilotage « manuelle » pour des tests simples. Par ailleurs, une simulation de processus peut s'interfacer avec un nombre virtuellement illimité de cartes (Figure 4.1).

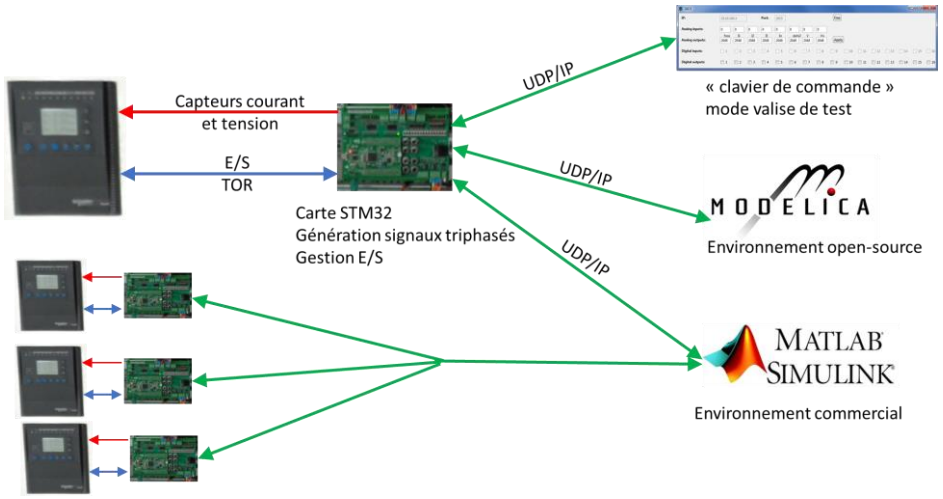


Figure 4.1 Principe de simulation Hardware-in-the-Loop de la plateforme G-ICS

Le système n'a pas de vocation temps réel[20] mais il permet de simuler des systèmes industriels ou des réseaux électriques avec une fidélité suffisante aux tests de cybersécurité. La plateforme a été utilisée pour la partie expérimentale des thèses de Maëlle Kabir-Querrec ([75]) et de Oualid Koucham.

Dans un autre volet de recherche, Ahmed Altaher a développé dans sa thèse [76] une approche fiabiliste pour l'évaluation de la dégradation des performances des fonctions de protection (temps de réponse en

[18] G-ICS : GreEn-ER Industrial Control systems Sandbox. GreEn-ER : Grenoble énergie - enseignement et recherche est un pôle d'innovation sur l'énergie et les ressources renouvelables
[19] Ecole Nationale Supérieure de l'Energie, l'Eau et l'Environnement
[20] Un system HIL temps réel commercial pour les réseaux électriques coûte environ 40000€ pour simuler un point de mesure (un relais de protection). La solution GICS revient à environ 400€ par équipement.

particulier) suite à des défaillances des équipements ou des dégradations de la disponibilité du réseau de communication ([77], [78], [79], [80]). Les modèles basés sur des bloc-diagramme de fiabilité et des réseaux bayésiens ont été validés avec des mesures expérimentales sur la plateforme. Ainsi, ont été menées par exemple, des campagnes de mesures de la dégradation du temps de réponse des fonctions de protection versus l'occupation de la bande passante du réseau [76].

La plateforme sert aussi de terrain expérimental pour les recherches menées au sein du Grenoble-Alpes Cybersecurity Institue[21], a fait l'objet d'une démonstration de scénarios attaque/défense au FIC 2018 et sera aussi présente au FIC 2019.

Perspectives

L'activité de développement de la plateforme est indispensable à l'ensemble de mes recherches en cybersécurité des systèmes industriels. Une des ambitions est de créer un nombre de cas d'étude des systèmes industriels dont les spécifications, les programmes des automates et les captures de trafic seront rendues publiques. Nous avons un premier cas d'étude très complet (Annexe A) associé aux recherches de la thèse de Oualid Koucham. Les jeux de données utilisés pour la fouille des spécifications et pour la détection ont été publiés [81] sur la plateforme grenobloise de partage de jeux de données PerSCiDo[22].

Un deuxième projet lié à la plateforme consiste à développer un environnement de test des systèmes de grande taille. En condition de laboratoire il est très difficile de dépasser un certain seuil quantitatif de matériel disponible. Par conséquent, afin de pouvoir reproduire des systèmes de grande taille, il est nécessaire de coupler le monde réel (IED et infrastructure réseau) avec un environnement virtuel (infrastructure réseau virtuelle, IED et automates virtualisés). Un simulateur libre de réseau de communication sera utilisé afin de garder un coût faible de réalisation. Le choix à priori se dirige vers ns-nam[23].

[21] Projet de recherche interdisciplinaire soutenu par l'IdEx grenoblois https://edu.univ-grenoble-alpes.fr/cybersecurity-institute-752300.htm
[22] https://persyval-platform.univ-grenoble-alpes.fr/0/searchbyrecently
[23] https://www.nsnam.org/

5 Nouvelles directions de recherche

Quitte à le répéter, un constat s'impose concernant la cybersécurité des systèmes industriels : aujourd'hui il est impossible de sécuriser un équipement industriel individuellement. Presque 5 ans après la Loi de la Programmation Militaire obligeant les Opérateurs d'Importance Vitale à prendre en compte explicitement la cybersécurité, seulement deux automates programmables ont passé la certification de premier niveau à court terme de l'ANSSI sous condition de désactivation des services de synchronisation du temps (NTP/SNTP), d'inventaire des biens (SNMP) et dans des configurations non-interopérables. Aucun SCADA n'a été certifié. Tant que la synchronisation du temps (indispensable dans les réseaux électriques) ne pourra être sécurisée, la certification sera impossible pour les IED. Des progrès plus importants ont été faits au niveau de l'authentification des images des systèmes d'exploitation embarqués et des connexions avec les environnements de développement. Au risque d'être polémique, j'affirmerai qu'après 5 ans nous sommes à la moitié de la première marche de la partie la plus facile du chemin menant à la sécurisation des équipements de contrôle commande.

Bien évidemment, il n'est pas question de remettre en cause le processus de sécurisation et certification des équipements. Il s'agit plutôt de prendre conscience de sa durée et du fait que le déploiement des futures architectures sécurisées va s'appliquer aux nouvelles installations, alors que la plus grande partie des installations existantes ne pourront pas être modernisées en raison des coûts prohibitifs de la réingénierie.

La tendance industrielle actuelle est de renforcer la carapace. Des bastions d'authentification à vocation industrielle pour implémenter le contrôle d'accès par rôles (Role-Based Acces Control), commencent à être proposés par les constructeurs d'automatismes industriels. C'est certainement une bonne nouvelle, mais renforcer uniquement le contrôle d'accès est insuffisant comme cela a été prouvé par toutes les attaques récentes sur des systèmes industriels. D'une part, la moindre vulnérabilité dans le contrôle d'accès met la menace face à un système sans défense, d'autre part, plusieurs statistiques montrent que plus de la moitié des intrusions sont réalisées par les menaces internes[24].

Il est donc nécessaire d'adopter une vraie vision de la défense en profondeur des systèmes industriels. En tant qu'automaticien je peux comprendre la réticence des exploitants quant au déploiement des contre-mesures cyber au sein des systèmes industriels. Certainement le plus problématique est le déploiement d'un IPS dans une infrastructure critique. La peur de voir une action légitime bloquée par l'IPS avec des possibles conséquences catastrophiques, est contre-argument principal. En effet, autant en régime normal le trafic réseau dans un système industriel est caractérisé par sa régularité, autant en régime d'urgence (dysfonctionnement ou perturbation du processus physique ou des actionneurs) des flux uniques (remontées des alarmes ou envoi des commandes d'arrêt d'urgence) seront envoyés. Un IPS qui bloque un flux car jamais observé auparavant, ou en raison d'une exception dans la syntaxe (requête propriétaire, par exemple) peut en effet déclencher des incidents industriels. Bien que moins grave, une remontée importante des fausses alertes par un IDS, éventuellement accompagnées par des message « obscurs » du point de vue de l'opérateur qui, ne l'oublions pas, ne sera pas un informaticien, mènera mécaniquement à terme à l'abandon de l'outil par l'exploitant.

Bien évidement ce ne sont pas des arguments pour abandonner l'étude de la cybersécurité des infrastructures critiques et se limiter au renforcement du contrôle d'accès. Les conclusions correctes sont les suivantes : il faut développer des outils performants et compréhensibles pas les exploitants des systèmes industriels et il faut accompagner le déploiement des contre-mesures avec le changement de la culture des automaticiens et ingénieurs électriciens.

En tant qu'enseignant dans une école d'ingénieurs en automatique et génie électrique, j'essaye de sensibiliser les futurs ingénieurs aux nouveaux risques cyber. En tant que chercheur en cybersécurité des systèmes industriels, j'essaie de porter un programme de développement des outils de recherche des

[24] Par exemple http://www.experian.com/data-breach/2016-ponemon-insider-risk.html

vulnérabilités, détection des intrusions et réaction, combinant la connaissance des systèmes industriels et de la sécurité informatique. Ma démarche est motivée par une caractéristique commune des menaces industrielles médiatisées[25] : à chaque fois la mise en œuvre de la menace a nécessité des connaissances approfondies du domaine industriels visé. Pour détecter de telles menaces une approche combinant les domaine physique et cyber me semble indispensable.

Les recherches précédentes en cybersécurité (thèses Kabir-Querrec et Koucham) ont visé principalement des aspects liés à la détection des intrusions et quelques prospectives sur la réaction face aux attaques (l'architecture résiliente de la Section 3.4.1.3). Outre les perspectives de recherche mentionnées en Sections 2.4 et 3.5, deux autres thématiques font partie de mon programme de recherche.

5.1 Recherche des vulnérabilités par rétroingénierie des logiciels embarqués

La rétroingénierie des logiciels embarqués aura deux objectifs :

1) Reconstituer le fonctionnement de la logique de contrôle par observation des comportements entrées/sortie.
2) Analyser le code embarqué, en particulier des bibliothèques de fonctions et des implémentations des protocoles, par des techniques statiques et dynamiques d'analyse de code binaire et des traces d'exécution.

L'idée sous-jacente au premier objectif est de comprendre ce qu'un attaquant en phase de reconnaissance pourrait obtenir comme informations sur le fonctionnement du système s'il n'a pas accès aux programmes des automates. Nous l'avons vu, une attaque orientée processus a un impact maximal si elle est déclenchée « au bon moment » c'est-à-dire dans l'étape adéquate du SFC. En reconstruisant le fonctionnement de la logique de contrôle, nous pouvons identifier les attaques possibles sur le système. Une autre retombée possible serait de proposer une manière de rendre obscure au moins une partie des états du SFC.

Le deuxième objectif trouve sa motivation dans le fait que le code embarqué sur les automates programmables ou les IED est généré automatiquement à partir d'un langage graphique de haut niveau (SFC, LD, FDB, CFC, ST, IL). Si on regarde les éléments de ces langages, on observe que très peu d'opérateurs sont disponibles (pratiquement que les opérateurs booléens font partie des langages). Les opérateurs arithmétiques (+ ,-,*,/,%,^, :=) sont implémentés sous forme de fonctions de bibliothèque (par exemple ADD, SUB, MIL, DIV, MOD, EXPT, MOVE) pour les opérations arithmétiques. La norme 61131-3 définit une centaine des fonctions de bibliothèque, dont la plus grande partie sont surchargées. Notre objectif est de spécifier des outils pour la vérification automatique du code binaire des implémentations et de recherche automatique des vulnérabilités exploitables par les menaces. Il s'agit par exemple des comportements inattendus lors de l'utilisation des combinaisons anormales des paramètres d'entrée. Dans une deuxième étape on aimerait développer le même type d'outils pour la vérification des briques élémentaires du code dans les IED (c'est-à-dire les implémentations des nœuds logiques standard).

Ce travail de recherche sera fait en collaboration avec Laurent Mounier du laboratoire Verimag et en nous 'appuyant sur des outils et techniques existants d'analyse de code ([82], [83], [84]).

5.2 Calcul autonomique et cyberdéfense des infrastructures critiques

L'idée de reconfiguration du système en cas d'attaque proposée comme une piste d'amélioration de la résilience du système dans la thèse de Maëlle Kabir-Querrec, nécessite un formalisme plus global. L'approche autonomique semble l'approche la plus adéquate.

L'autonomique est un concept relativement nouveau dans l'informatique. En [85] les auteurs proposent un modèle de boucle de rétroaction associée au concept d'auto-gestion des systèmes informatiques.

[25] De Maroochy Shire à Industroyer/CrashOverride en passent par Stuxnet et BlackEnergy

L'idée sous-jacente est que la reconfiguration dynamique des systèmes informatiques devient de plus en plus difficile à gérer étant donné la complexité croissante des logiciels et systèmes informatiques. Un système informatique autonomique présente quatre propriétés principales qu'il est censé maintenir de manière autonome : l'auto-configuration (la capacité d'apprendre et s'adapter), l'auto-protection face aux menaces externes, l'auto-réparation et l'auto-optimisation.

La boucle de rétroaction autonomique (Mape-K - Monitor-Analyze-Plan-Execute over a shared Knowledge -Figure 5.1) est un modèle de référence pour le contrôle des systèmes autonomiques.

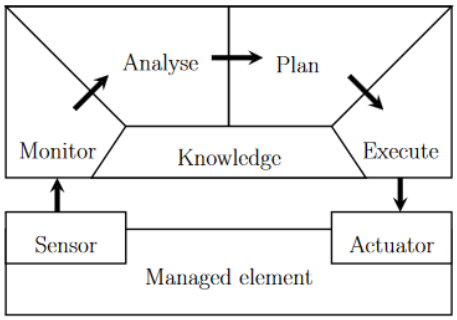


Figure 5.1 Boucle de rétroaction autonomique MAPE-K

L'approche autonomique et le modèle MAPE-K ont été utilisés en divers domaines, dont le Cloud-Computing, Smart-Grids, Smart-Cities mais aussi dans la cybersécurité des systèmes informatiques. Dans le domaine de la cybersécurité des systèmes industriels quelques travaux ont été publiés ([86], [87], [88]) et même une thèse [89]. La majorité des travaux cités dans le domaine des SCADA proposent soit un IDS soit un IPS auto-apprenant de nouvelles règles de détection/protection à partir du monitoring du trafic réseau et d'un certain nombre de connaissances.

Une exception notable est l'approche présentée en [90] qui propose une reconfiguration via OpenFlow, en cas d'attaque, du réseau de communication du smart-grid afin d'isoler la partie du réseau compromise. C'est effectivement un exemple de reconfiguration du système au niveau cyber afin d'améliorer sa résilience.

D'autres travaux récents proposent de formalismes qui prennent en compte l'approche autonomique pour la détection [91], [92] ou l'opacité (l'incapacité de reconnaitre l'état interne du processus) [93], [94].

L'approche que je propose pousse la reconfiguration du système au jusqu'au niveau physique. Ainsi, en cas d'attaque, le système auto-reconfigure le réseau de communication (niveau cyber) et les modes de fonctionnement des automates programmables et/ou IED et SCADA afin de préserver un niveau de service défini. Pour un réseau électrique il s'agit de maximiser le nombre de consommateurs alimentés avec les équipements non-compromis. Dans le cas d'un système d'automatisme industriel, il s'agit de maintenir les fonctions de sécurité du système et chercher un mode de repli en s'appuyant sur les équipements disponibles.

L'autonomique représente le domaine d'activité principal de mon équipe INRIA/LIG (Ctrl-A). Cette thématique d'auto-protection des systèmes industriels face aux menaces cyber démarrera dès cette année.

Notons que la boucle MAPE-K peut elle-même être regardée comme un contrôleur et des méthodes de synthèse de l'automatique peuvent être appliquées [91].

6 Autres travaux de recherche

La cybersécurité est mon principal domaine d'activité de recherche depuis 2013. Précédemment j'ai travaillé dans le domaine des systèmes stochastiques. Mon sujet de thèse portait sur les propriétés théoriques des chaînes de Markov absorbantes [91], [92], [93]. Ces recherches s'encadraient dans la thématique évaluation des performances des systèmes à évènements discrets développée au sein du Laboratoire d'Automatique de Grenoble. Durant mon post-doc à Paris j'ai commencé des recherches sur les méthodes approximatives de résolution de réseaux de files d'attente fluides. Une file d'attente fluide est une approximation d'une file d'attente discrète avec arrivées et traitements en rafales (Figure 6.1).

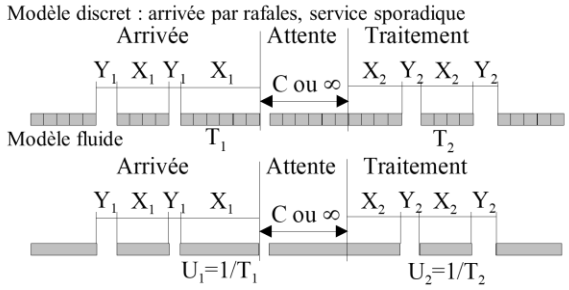


Figure 6.1 Modèle discret en rafales et son approximation fluide

Ces types de modèles sont utilisés en évaluation des performances pour les lignes de production manufacturières, mais aussi pour les réseaux de communication, et particulièrement pour les réseaux ATM. Le processus stochastique correspondant aux réseaux des files d'attente fluides est appellé un processus Markovien déterministe par morceaux (Piece-wise Deterministic Markov Process - PDMP). En dehors des applications d'évaluation des performances, les processus PDM ont été utilisés dans des modèles de fiabilité, mais aussi dans l'automatique pour la modélisation des systèmes linéaires à commutation stochastique.

Pendant plusieurs années j'ai développé des applications d'évaluation de performances basées sur des PDMP pour les systèmes manufacturiers mais aussi pour les réseaux de communication (thèse Alexandre Royer [94]). En parallèle j'ai travaillé sur l'étude stochastique des systèmes de fabrication complexes avec des flux réentrants (applications dans l'industrie des semiconducteurs – thèse Hai Binh Nguyen [95]).

Après la restructuration des laboratoires de recherche en 2006, la thématique systèmes à évènements discrets a pratiquement été arrêtée dans le laboratoire nouvellement créé (Gipsa-lab) à la suite du départ de la plus grande partie de chercheurs vers le génie industriel. En 2008 j'étais le seul chercheur permanent à travailler encore dans l'évaluation des performances.

En continuité des recherches sur les PDMP j'ai travaillé jusqu'en 2013 sur la commande optimale des systèmes linéaires à commutation stochastique (thèse Simona Mihaita [96], [97]).

Je n'ai pas abandonné l'activité « stochastique » bien que la quasi-totalité de mes recherches se situe dans le domaine de la cybersécurité. Actuellement je participe au co-encadrement d'une thèse localisée à Brest (Chaba Hireche) sur la partie planning optimal, par des modèles de décision markovienne, de la mission d'un drone.

7 Synthèse des activités et responsabilités.

7.1 Responsabilités administratives et d'enseignement

- Responsable de la salle de Travaux Pratiques Réseaux d'ordinateurs de l'ENSE3 (ENSIEG) de 2000 à 2014
- Depuis 2014 : responsable de la plate-forme de cybersécurité et communication industrielle G-ICS (commune Grenoble-INP et Université Grenoble Alpes)
- Septembre 2010 - septembre 2012 Responsable parcours de formation 3A Conduite et Supervision (12 à 18 étudiants) de l'ENSE3
- Septembre 2012 - décembre 2013 Responsable 2A filière de formation Automatique Systèmes et Information (ASI) de l'ENSE3 (environ 48 étudiants)
- Janvier 2014 – juillet 2017 : Responsable filière de formation ASI de l'ENSE3 et 2A.
- Septembre 2014 – aujourd'hui : Responsable de trois Unités d'Enseignement (U.E.) à l'ENSE3
- Depuis septembre 2017 :
  - Charge de Mission « Transition Numérique » à l'ENSE3
  - Co-pilote du processus « Support » dans le cadre de la démarche d'amélioration continue de l'ENSE3
- Porteur de la demande de labellisation CyberEDU de l'ENSE3

7.2 Responsabilités de contrats de recherche

- Programme PULSE de l'IRT Nanoelec 2018. Je suis leader de deux workpackages dotées d'un budget de 150k€ dont 3 postes (un post-doc et deux poste ingénieur support)
- Grenoble Alpes Cybersecurity Institute ¨(GACI) 2017. Je suis co-porteur du work-package WP3 Methodology for vulnerability analysis and certification. Le budget total du projet est de 1,4M€.
- ANR HPEC édition 2015 CE24. Je suis l'unique participant (et donc responsable scientifique et technique) du partenaire GIPSA-lab. Aide ANR de 15600€ (missions et équipement) pour un coût total de 51k€ (équivalent à 6 mois d'enseignant-chercheur)
- ANR SACADE édition 2016 ASTRID (financement DGA). Je suis responsable scientifique du partenaire GIPSA-lab (2 permanents et 3 thésards). Aide ANR/DGA de 88k€ (matériel et 12 mois ingénieur) pour un coût total de 416k€. Du point de vue de l'aide accordée je représente le partenaire le plus important du consortium.
- Contrat de subvention CIFRE DGA (thèse Koucham). Co-responsable avec Jean-Marc Thiriet. Montant : 147k€ (salaire du doctorant inclus). 2014-2017
- Contrat d'accompagnement CIFRE EuroSystem (thèse Kabir-Querrec). Co-responsable avec Jean-Marc Thiriet. Montant 30k€. 2013-2016
- Contrat de recherche INCAM Solutions (groupe 40-30). Co-responsable avec Christian Commault, 10k€. 2003
- Contrat de recherche Prodipact. Responsable. 10k€. 2002

7.3 Responsabilités projets d'enseignement

La plus grande partie des projets d'enseignement dont je suis responsable sont lié au développement de la plateforme enseignement-recherche G-ICS (GreEn-ER Industrial Control systems Sandbox) financée majoritairement sur des projets d'enseignement :

- Appel au projets plateformes de Grenoble-INP 2014 : 100k€
- Appels aux projets plateformes du Labex grenoblois Persyval-lab 2013/2014/2015/2016 : 120k€
- Appel des projets plateformes « learning-by-doing » de l'IdEx UGA 2016 : 60k€
- Appel aux projets IRT Nanoelec formation 2017 : 48k€

Autres projets : responsable du projet formation l'IdEx de l'Université Grenoble-Alpes TransNumE3, budget 150k€

Récapitulatif des thèses encadrées

Alexandre Royer. Evaluation de performances de réseaux de communication à l'aide de chaînes de Markov hybrides

- Début : septembre 2002, Soutenue : janvier 2006
- Encadrement : Stéphane Mocanu 50%, Christian Commault 50%
- Publications :
  - Alexandre Royer, Stéphane Mocanu, Christian Commault. Méthode de décomposition pour l'évaluation de performances de réseaux linéaires de routeurs ON/OFF, *Revue des Sciences et Technologies de l'Information - Série TSI : Technique et Science Informatiques*, Lavoisier, 2005, pp.179-202 < hal-00385037v1>
  - Alexandre Royer, Stéphane Mocanu, Christian Commault, A decomposition method to evaluate the performance of fluid linear networks with identical ON/OFF routers. 8ème Atelier d'evaluation de performances, Reims, 2003
- Alexandre Royer est actuellement ingénieur Alcatel

Hai Binh Nguyen : Politiques de pilotage pour l'optimisation d'un système de production de semi-conducteurs

- Début : septembre 2004, Soutenue décembre 2007
- Encadrement : Stéphane Mocanu 50%, Christian Commault 50%
- Publications :
  - Hai Binh Nguyen, Christian. Commault, and Stéphane. Mocanu. Etude en simulation de politiques de pilotage multiproduit dans l'industrie des semi-conducteurs. In Colloque IPI : "Comprendre et piloter la mutation des systèmes de production", 2006 <hal-00198461v1> publié en chapitre d'ouvrage en J.F. Boujut, D. Llerena, D. Brissaud. *Les systèmes de production : applications interdisciplinaires et mutations*, Hermès sciences publications, pp.33-44, 2007
  - Hai Binh Nguyen. Politiques de pilotage de systèmes à flux complexe, *2èmes Journées Doctorales/Journées Nationales MACS 2007 (JD-JN-MACS 2007)*, Jul 2007, Reims, France. <hal-00154415v1>
- Hai Binh Nguyen est ingénieur en Suisse.

Adriana Simona Mihaita : Approche probabiliste pour la commande orientée évènement des systèmes stochastiques à commutation

- Début : septembre 2009, Soutenue septembre 2012
- Encadrement : Stéphane Mocanu 90%, Hassane Alla 10%
- Publications :
  - Adriana Simona Mihaita, Stéphane Mocanu, Pascal Lhoste, , "Probabilistic analysis of a class of continuous-time stochastic switching systems with event-driven control", *European Journal of Automation* (accepted July 2016).
  - Adriana Simona Mihaita, Stéphane Mocanu. Un nouveau modèle de l'énergie de commande des systèmes stochastiques à commutation, *Septième Conférence Internationale Francophone d'Automatique (CIFA 2012)*, Jul 2012, Grenoble, France. pp.WePM4T9.2, 2012 < hal-00739300v1>
  - Adriana Simona Mihaita, Stéphane Mocanu. Simulation à événements discrets en temps continu pour la commande orientée événements des systèmes stochastiques à commutation, *MSR 2011* Lille, publié en *Journal Européen des Systèmes Automatisés (JESA)*, Lavoisier, 2011, 45 (1-3), pp.157-172. <10.3166/jesa.45.157-172>< hal-00630164v1>

    - Adriana Simona Mihaita, Stéphane Mocanu. An energy model for event-based control of a switched Integrator, *18th IFAC World Congress (IFAC WC 2011)*, Aug 2011, Milan, Italy. pp.2413-2418, 2011 <<hal-00630155v1>
- Adriana Simona Mihaita est chercheuse CSIRO (Commonwealth Scientific and Industrial Research Organisation) à Sidney en Australie.

Maëlle Kabir-Querrec : Cybersécurité des systèmes de contrôle pour les smart-grids

- Date d'entrée en thèse 01/10/2013, Soutenue juin 2017
- Encadrement : Stéphane Mocanu 60% , Jean-Marc Thiriet 40%
- Publications :
    - Maëlle Kabir-Querrec, Stéphane Mocanu, Jean-Marc Thiriet, Eric Savary. A Test bed dedicated to the Study of Vulnerabilities in IEC 61850 Power Utility Automation Networks. *21st IEEE Emerging Technologies and Factory Automation, (ETFA 2016)*, Berlin, Germany, September 2016, 2016, <http://www.etfa2016.org/index.php>. <hal-01366270>
    - Maëlle Kabir-Querrec, Stéphane Mocanu, Pascal Bellemain, Jean-Marc Thiriet, Eric Savary. Architecture des systèmes d'automatisation des postes résiliente aux attaques des trames GOOSE. *Journées C&ESAR 2015*, Nov 2015, Rennes, France. 2015, <http://www.cesar-conference.org/>. <hal-01237722>
    - Maëlle Kabir-Querrec, Stéphane Mocanu, Pascal Bellemain, Jean-Marc Thiriet, Eric Savary. Corrupted GOOSE Detectors: Anomaly Detection in Power Utility Real-Time Ethernet Communications. *GreHack 2015*, Nov 2015, Grenoble, France. <hal-01237725>
    - Maëlle Kabir-Querrec, Stéphane Mocanu, Jean-Marc Thiriet, Eric Savary. Power Utility Automation Cybersecurity: IEC 61850 Specification of an Intrusion Detection Function. *25th European Safety and Reliability Conference (ESREL 2015)*, Sep 2015, Zürich, Switzerland. <hal-01237713>
    - Stéphane Mocanu, Maëlle Kabir-Querrec, Jean-Marc Thiriet, Eric Savary. Cybersécurité des sous-stations électriques IEC 61850: Présentation de travaux de thèse. *Rendez-Vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information (RESSI 2015)*, May 2015, Troyes, France. <hal-01237730>
- Maëlle Kabir-Querrec est Full Senior Researcher chez ABB Suisse.

Ahmed Altaher : Mise en œuvre d'un cadre de sureté de fonctionnement pour les systèmes de contrôle industriel : application à des systèmes de distribution d'énergie électrique (smart grids)

- Date d'entrée en thèse avril 2013 (interruption pour des raisons de santé et médicales de plus de 8 mois. Soutenue février 2018
- Encadrement : Stéphane Mocanu 50%, Jean-Marc Thiriet 50%
- Publications :
    - Ahmed Altaher, Stéphane Mocanu, Jean-Marc Thiriet. Dependability Optimization of Process-level Protection in an IEC-61850-Based Substation. Walls, Revie & Bedford. *26th European Safety and Reliability Conference (ESREL 2016)*, Sep 2016, Glasgow, United Kingdom. Tayor & Francis Group, *Risk, Reliability and Safety: Innovation Theory and Practices - ESREL 2016*, pp.284, 2016,. <hal-01380261>
    - Ahmed Altaher, Stéphane Mocanu, Jean-Marc Thiriet. Evaluation of Time-Critical Communications for IEC 61850-Substation Network Architecture. *Surveillance 8 International Conference* , Oct 2015, Roanne, France. Proceeding of Surveillance 8 2015, <http://surveillance8.sciencesconf.org/>. <hal-01242297>
    - Ahmed Altaher, Stéphane Mocanu, Jean-Marc Thiriet. Experimental Evaluation of an IEC 61850-Station Bus Communication Reliability. *Journées Nationales des*

Communications Terrestres (JNCT 2015), Jun 2015, Valence, France. 2015. <hal-01242280>

- Ahmed Altaher, Jean-Marc Thiriet, Stéphane Mocanu. Performance Evaluation of IEC 61850 safety-related communication architecture. *Summer School on Cyber-Physical Systems - 2014* Edition, Jul 2014, Grenoble, France., <hal-01242259>

- Ahmed Altaher a obtenu un poste à l'Université de Tripoli

Oualid Koucham : Détection d'intrusions dans les systèmes de contrôle industriels

- Encadrement : Stéphane Mocanu 40%, Guillaume Hiet (CentraleSupelec Rennes) 40%, Jean-Marc Thiriet 20%
- Date d'entrée en thèse 01/10/2015, soutenance prévue novembre 2018
- Publications :
  - Oualid Koucham, Stephane Mocanu, Guillaume Hiet, Jean-Marc Thiriet, Frédéric Majorczyk, Efficient Mining of Temporal Safety Properties for Intrusion Detection in Industrial Control Systems,Conference: Accepté à 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, 2018
  - Oualid Koucham, Stéphane Mocanu, Guillaume Hiet, Jean-Marc Thiriet, Frédéric Majorczyk. Detecting Process-Aware Attacks in Sequential Control Systems, 21st Nordic Conference on Secure IT Systems (NordSec 2016), Nov 2016, Oulu, Finland. http://nordsec.oulu.fi <hal-01361081v1>

Chabha Hireche (thèse localisée à UBO à Brest), Etude et l'implantation au sein d'un système sur puce, d'une approche probabiliste de contrôle de mission de drone autonome.

- Encadrement : Jean-Philippe Diguet 40%, Catherine Dezan 50%, Stéphane Mocanu 10%
- Date d'entrée en thèse 01/11/2015, soutenance prévue novembre 2018
- Publication :
  - Chabha Hireche, Catherine Dezan, Jean-Philippe Diguet, Stéphane Mocanu. Planification de Mission de Drone: Implémentation Logicielle/Matérielle. *GDR SoC2*, Jun 2018, Paris, France

A. Description du processus de test

Nous décrivons par la suite le scénario d'usage utilisé pour la partie expérimentale et l'évaluation des résultats de la thèse d'Oualid Koucham. Cela constitue le premier cas public d'étude déployé sur la plateforme GICS. Le processus est dérivé d'un cas d'étude classique en automatique.

Le processus chimique dit « Tennessee Eastman » est utilisé depuis plus de 25 ans comme problème type pour l'évaluation des stratégies de commande, de diagnostic et plus récemment de sécurité. Depuis la première publication des spécifications du processus chimique par des ingénieurs de Eastmann Chemical Company en1993, plusieurs versions et adaptations du processus ont été proposées. Nous utilisons ici une adaptation du processus original.

Description du processus

L'objectif de l'installation chimique (voir Figure A.1) est la synthèse d'un produit à partir d'une réaction chimique incluant deux réactifs. L'installation s'inspire du procédé de désalkylation du toluène en benzène. Le procédé se déroule selon plusieurs sous-processus incluant la synthèse de deux réactifs et la réaction chimique proprement dite. La réaction chimique n'étant pas complète, une étape de séparation et de distillation permet de recycler les réactifs non consommés.

Le premier réactif est synthétisé dans le silo S1 à partir des produits initiaux P1 et P2. Pour ce faire, une quantité adéquate du produit initial P1 (resp. P2), indiquée par le capteur W1 (resp. W2), est introduite dans le réacteur TK1 via les vannes VP1 et VP2. Les produits introduits dans TK1 subissent une phase de mélange par l'intermédiaire du moteur M1 pour une durée de 50s. Le mélange est alors vidangé vers la trémie tampon TP1 via la vanne VT1. Ce mélange est ensuite vidangé vers le réacteur TK2 puis la production d'un nouveau mélange à partir de P1 et P2 commence si le silo S1 n'est pas encore totalement rempli. En parallèle, le produit dans TK2 est mélangé avec un autre produit P5 acheminé via le chariot CH1 (capteurs de fin de course SH1 et SB1). Le mélange se fait par l'intermédiaire du moteur M2 pour une durée de 60s. Le mélange est ensuite vidangé vers le silo S1 avant le début d'une nouvelle phase si la trémie tampon TP1 n'est pas vide. L'ensemble des étapes précédentes se déroulent de façon à garantir que le silo S1 n'est jamais vide (capteur NB1).

Le deuxième réactif est synthétisé dans le silo *S2* à partir des produits initiaux *P3* et *P4* de manière analogue au premier réactif. La principale différence réside dans une phase de test d'un échantillon du réactif dans le réacteur *TK4*. Après la phase de mélange dans *TK4*, un échantillon du réactif est prélevé via la vanne *VS1* vers *QC1*. Le résultat du test est renvoyé par le capteur *SMPL*. Dans le cas d'un résultat négatif, le réactif dans *TK4* est vidangé hors du processus via la vanne *VO*. Dans le cas contraire, le réactif est acheminé vers le silo *S2*.

Une quantité adéquate de réactifs est acheminée depuis les silos *S1* et *S2* vers le réservoir *S3*. Les quantités sont déterminées à l'aide des capteurs de niveau *NM3* et *NH3*. Les réactifs sont alors chauffés dans *HT1* avant d'être acheminés vers le réacteur *RE1* où se produit la réaction. Les produits de la réaction sont refroidis dans *CO1* puis sont séparés à l'aide des séparateurs *SP1*, *SP2* ainsi que de la colonne de distillation *DI1*. La séparation dans *SP2* et le procédé de distillation se déroulent en parallèle. Les produits inutiles *P8* et *P9* sont rejetés via les vannes *VO5* et *VO4*. Le produit final *P10* est recueilli depuis la colonne de distillation via la vanne *VO6*. Les réactifs résiduels sont recyclés vers les silos *S1* et *S2* via les vannes *VO3* et *VO7*.

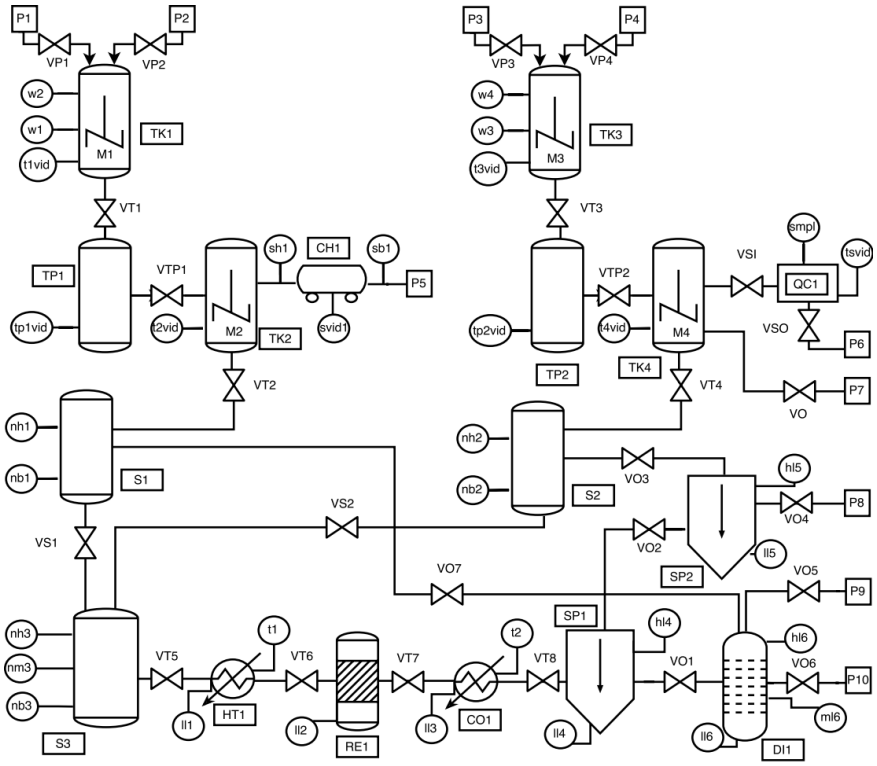


Figure A.1 Processus physique du cas d'étude usine chimique

Système de contrôle

Le contrôle du processus est réalisé par une architecture distribuée et hiérarchique. Le processus global est découpé au niveau de la commande en trois sous-processus. A l'intérieur de chaque sous-processus de boucles locales réalisent la régulation des composantes principales (séparateurs, réacteurs, colonne de distillation).

Découpage du processus, commande distribuée

Les trois sous-processus identifiés correspondent aux successions d'opérations nécessaires à la réalisation des mélanges P1/P2/P5 respectivement P3/P4/P7 ainsi qu'à la distillation finale des produits.

Le premier sous-processus est donc constitué de l'ensemble des composantes TK1, TP1, TK2, CH1, S1 ainsi que les capteurs et actionneurs associées. La gestion des opérations est gérée par un automate programmable M340 avec un réseau des RTU OTB 1C0 DM9LP sur bus CAN. La Figure A.2 reproduit le schéma de ce sous-processus.

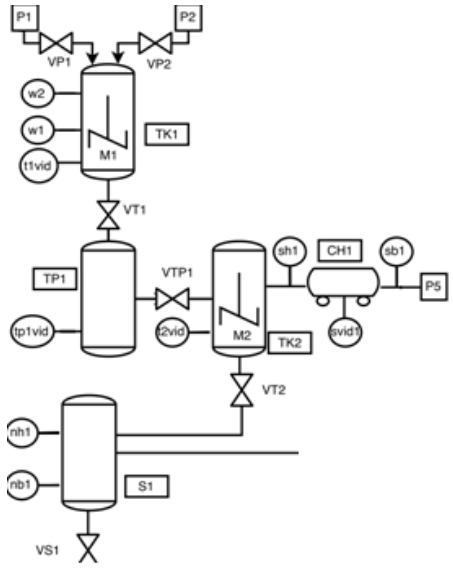


Figure A.2 . Sous-processus 1, mélanges P1/P2/P5

Le second sous-processus est constitué de la chaîne de réacteurs et réservoirs TK3/TP2/TK4/S2/QC1 ainsi que du dispositif de contrôle de qualité QC1. Le sous-processus est représenté en Figure A.3. Le contrôle du sous-processus est fait par un automate M580.

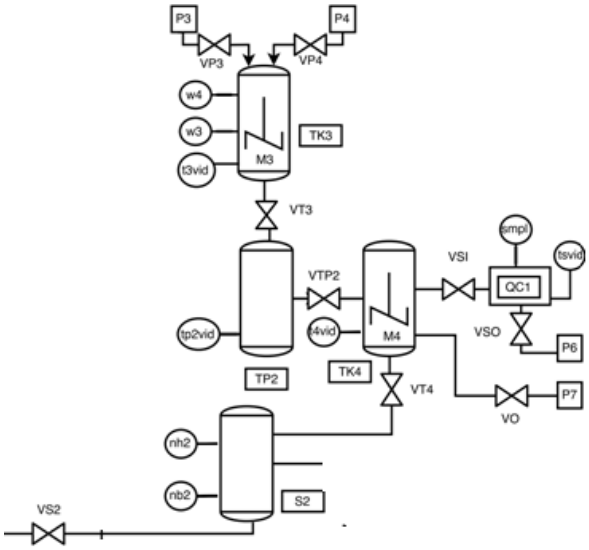


Figure A.3. Sous-processus 2, mélanges P3/P4/P7 et contrôle de qualité

Finalement le troisième sous-processus correspond à la phase de distillation et à ses opérations préliminaires et finales. Le sous-processus est piloté par un automate M580 avec un module RTU déporté sur Ethernet/IP. La Figure A.4 présente le schéma du dernier sous-processus.

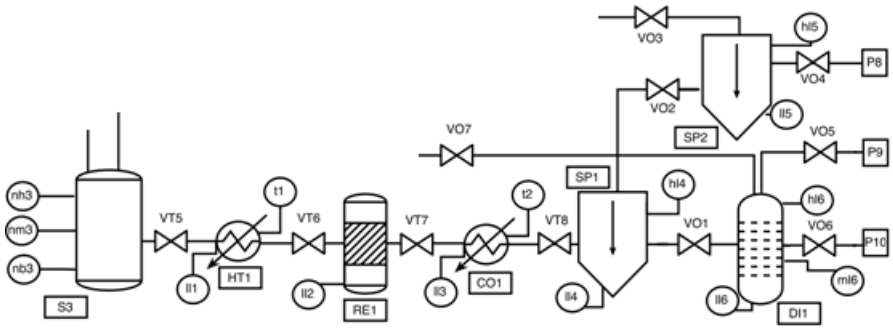


Figure A.4. Sous-processus 3 : mélange final en réacteur 3 et distillation

Chaque contrôleur peut être utilisé en deux modes : manuel et automatique. En mode automatique le programme de l'automate contrôle le processus. En mode manuel les commandes sont envoyées par un opérateur via un synoptique de supervision.

Du point de vue des entrées-sorties E/S du processus, au niveau du pilotage global, on compte 71 variables correspondant à des capteurs/actionneurs et trois variables de commande pour passage en mode manuel.

Le tableau suivant présente la répartition des variables par sous-processus :

Sous-processus	**Capteurs**	**Actionneurs**	**Autres**
Mélange 1	*W1,W2,T1VID,TP1VID, SB1,NB1,NH1,T2VID, SVID1,SH1*	*VP1,VP2,VTP1,VT2, VS1,M1,M2,MC1, D1,VT1*	*MANUAL*
Mélange 2	*W3,W4,T3VID,TP2VID, SMPL,NB2,NH2,T4VID, TSVID*	*VP3,VP4,VTP2,VT4, VS2,M3,M4,VSI, VSO,VT3,VO*	*MANUAL*
Mélange 3	*NM3,NH3,NB3,T1, LL1,LL2,T2,LL3, HL4,HL5,LL5,LL4, HL6,ML6,LL6*	*VT4,H1ON,VT6,R1ON, VT7,H2ON,VT8,VO2, VO5,VO1,VO4, VO3,VO6,VO7*	*MANUAL*

Tableau 4.1. Répartition des variables E/S et contrôle par sous-processus

Architecture des protocoles

Trois protocoles de communication interviennent dans la communication au niveau de la commande distribuée. Deux des protocoles sont IP : Modbus/TCP entre les automates et Ethernet/IP entre l'API de commande du sous-processus 3 et son RTU. Le troisième protocole est CAN sur bus de communication série RS-485. La Figure A.5 présente schématiquement l'architecture réseau. Les équipements d'interconnexion ne sont pas représentés. Les flèches bidirectionnelles représentent les flux de données : communications bidirectionnelles avec les RTU, ainsi qu'entre l'automate de commande du sous-processus 3 et ceux des sous-processus 1 et 2. Les sous-processus 1 et 2 étant réciproquement indépendants, il n'y a pas de flux de données entre leurs deux automates de commande.

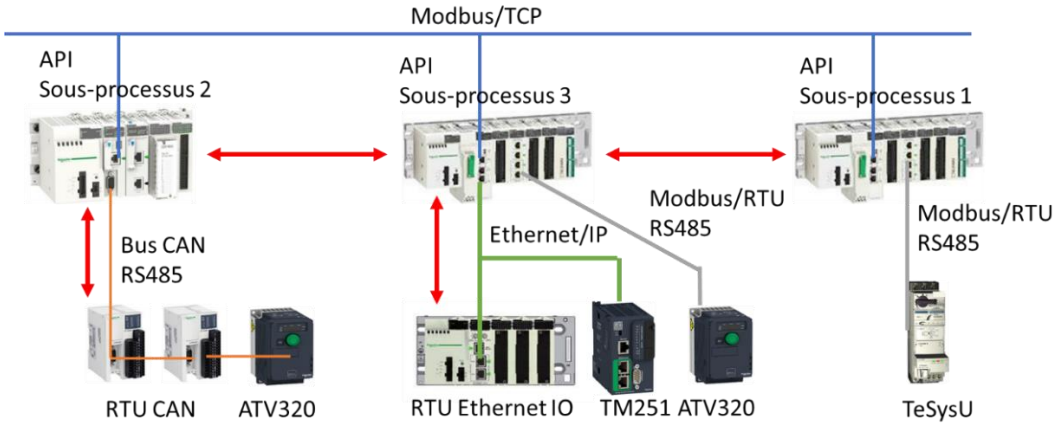


Figure A.5. Architecture des protocoles et flux de données dans le système de commande distribuée

Supervision et historisation

La supervision et l'acquisition des données sont implémentées à trois niveaux : niveau local par sous-processus, niveau distant (salle de contrôle) via des serveurs OPC et finalement historisation via une base de données SQL.

Supervision locale

Le but de la supervision locale est de permettre le suivi et la commande manuelle d'une partie du processus global. Une IHM Magelis communiquant sur un bus de terrain (CAN ou ModbusRTU) est déployée pour chaque commande locale de moteur et pour la commande de la colonne de distillation. Les sus-processus 1 et 2 sont également supervisés par des IHM Magelis communiquant via Modbus/TCP. Un synoptique plus général est déployé sous PCVue afin de superviser le sous-processus 3 et les variables des sous-processus 1 et 2 influant sur le sous-processus 3.

Supervision distante

Deux serveurs OPC sont déployés : un serveur OPC DA (OFS Factory server) et un serveur OPC-UA (Kepware). Le serveur OPC-DA matérialise la sale de commande du site, les clients se connectant au serveur OPC-DA dans le même réseau local. Le serveur OPC-UA permet des connexions de l'extérieur du site par des clients légers. Certaines commandes des clients peuvent être envoyées directement aux automates via le protocole Modbus/SOAP.

Historisation des donnés

Un serveur OPC est alimenté par le synoptique PCVue via des requêtes SQL. Des clients SQL peuvent se connecter de l'extérieur du site.

Synchronisation du temps.

Un serveur NTP local permet la synchronisation du temps des différents éléments du système.

Architecture de supervision

La Figure A.6 présente l'architecture et les protocoles de communication de la supervision. Les équipements d'interconnexion ne sont pas représentés.

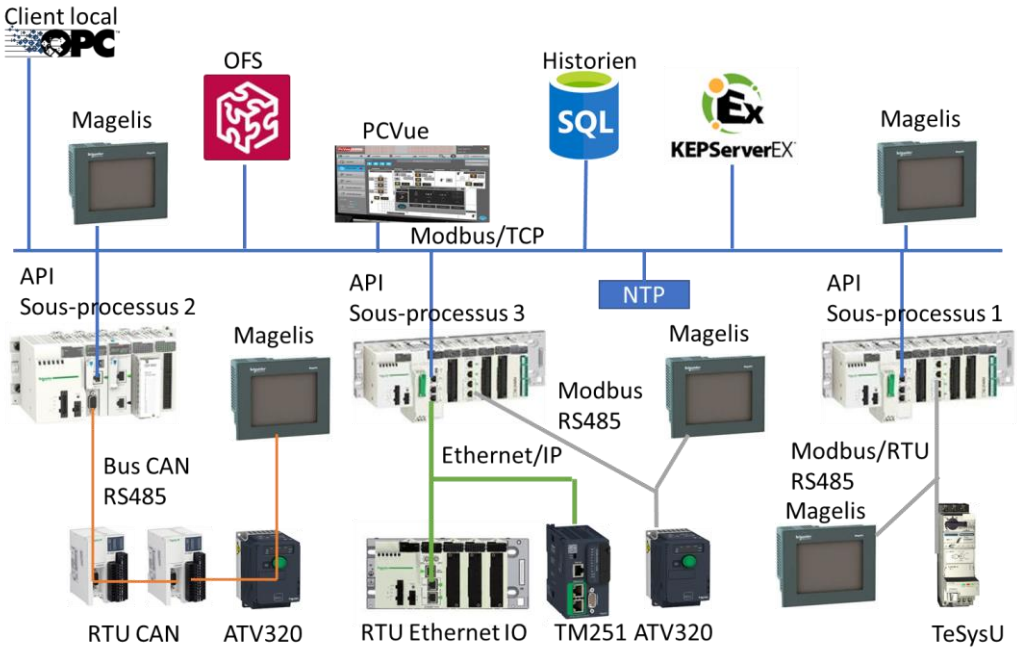


Figure A.6. Architecture de communication

Les flux de communication sont représentés en Figure A.7. Pour alléger le schéma, les noms des protocoles de commande et les noms des équipements ne sont plus représentés. Les flèches bidirectionnelles représentent les flux de données de supervision. Sauf indication contraire, les flux de données sont de type Modbus/TCP.

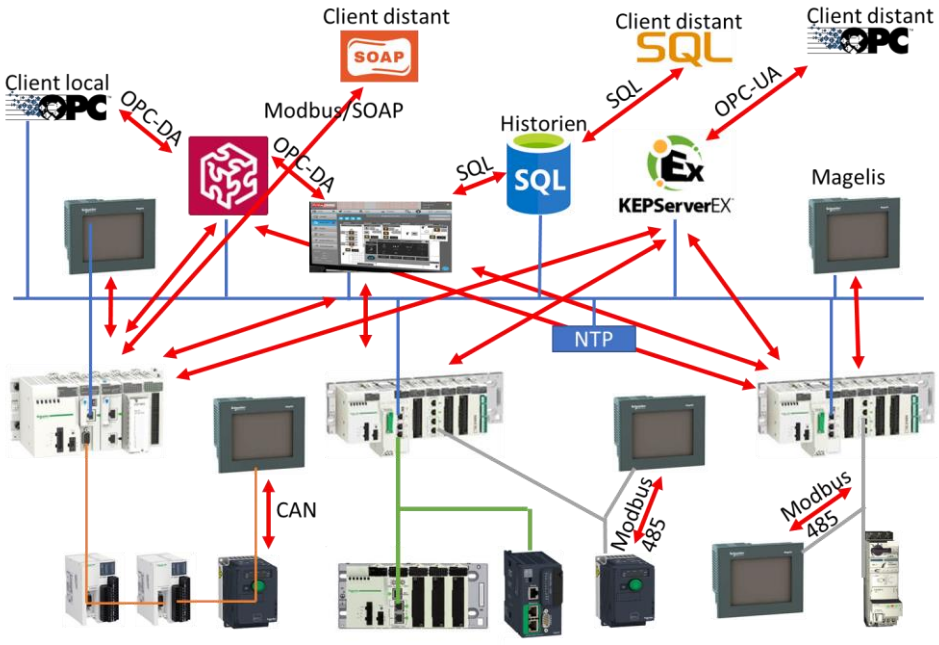


Figure A.7. Flux de communication de supervision

B. Expérimentation : attaques déployées et monitoring

Pour l'évaluation de la détection nous avons déployé 5 attaques quantitatives et 5 qualitatives (Tableau B-1). Le Tableau B-2 présente les moniteurs qui alertent chaque type d'attaque.

Attaque	Objectif	Variable Ciblée	Type
A1	Fausser les proportions des produits P1 et P2 en TK1	VP2@PLC1	Qualitatif
A2	Fausser les proportions des produits P1 et P2 en TK1	VP1@PLC1 VP2@PLC1	Qualitatif
A3	Usure du moteur M1 par séquences rapides start/stop	M1@PLC1	Quantitatif
A4	Transfert prématuré des produits en S1	VT2@PLC1	Qualitatif
A5	Raccourcissement de la durée du mélange en TK2	M2@PLC1	Quantitatif
A6	Fausser les proportions des produits P3 et P4 en TK2	VP4@PLC2	Qualitatif
A7	Usure du moteur M3 par séquences rapides start/sto	M3@PLC2	Quantitatif
A8	Raccourcissement de la durée du mélange en TK4	M4@PLC2	Quantitatif
A9	Transfert prématuré des produits en S2	VO@PLC2 VT4@PLC2	Qualitatif
A10	Vidange du réacteur durant la réaction	R1On@PLC3	Quantitatif

Tableau B-1 Liste des attaques et actionneurs ciblés

On observe que A4 viole deux propriétés de sécurité. En réalité il s'agit de la même propriété (pas d'ouverture de VT2 avant l'approvisionnement en réactants) qui a été identifiée de deux manières différentes (deux patrons qualitatifs différents) lors de la fouillée. L'ajour d'une connaissance technologique du processus permettrait de fusionner les deux spécifications.

L'attaque A9 manipule deux actionneurs pour vider TK4. Deux moniteurs LTL lèvent donc des alertes. L'attaque viole aussi une propriété qualitative sur la durée du mélange.

Ces phénomènes (propriétés de sécurité opératives caractérisées par plusieurs expressions ainsi que les attaques qui déclenchent des alertes quantitatives et qualitatives) nous ont incités à développer l'approche de corrélation.

Attaque	Propriété violée	
	Qualitative	Quantitative
A1	$(t1vid^{\downarrow}, p1^{\uparrow})$:absence(vp2)	
A2	$(p1^{\uparrow}, p2^{\uparrow})$:absence(vp2)	
A3		$(p2^{\uparrow}, p2^{\downarrow})$:durationMin(m1,50s)
A4	$(tp1vid^{\uparrow}, sb^{\downarrow})$:absence(vt2) $(sb^{\downarrow}, sh^{\uparrow})$:absence(vt2)	
A5		$(svid^{\downarrow}, act^{\downarrow})$:durationMin(m2,60s)
A6	$(t2vid^{\downarrow}, p3^{\uparrow})$:absence(vp4)	
A7		$(p4^{\uparrow}, p4^{\downarrow})$:durationMin(m3,50s)
A8		$(tsvid^{\downarrow}, act^{\downarrow})$:durationMin(m4,60s)
A9	$(tsvid^{\uparrow}, act^{\downarrow})$:absence(vt4) $(tsvid^{\uparrow}, act^{\downarrow})$:universality(vo)	$(tsvid^{\uparrow}, act^{\downarrow})$:durationMin(vo,30s)
A10		$(ll1^{\downarrow}, ll3^{\uparrow})$:durationMin(r1on,400s)

Tableau B-2 Propriétés de sécurité violées par les différents types d'attaques.

Bibliographie

[1] ANSSI, «Maîtriser la SSI pour les systèmes de contrôle industriels,» ANSSI, 2012.

[2] ANSSI, «La cybersécurité des systèmes industriels: Méthode de classification,» ANSSI, 2012.

[3] V. Hunt, «The Need for Computer-Integrated Manufacturing.,» chez *Computer Integrated Manufacturing Handbook.* , 1989, pp. 3-44.

[4] K. Stouffer, J. Falco et K. Scarfone, «SP 800-82 Rev 2. Guide to Industrial Control Systems (ICS) Security,» NIST, 2015.

[5] CEI 62443, *Cybérsécurité des installations industrielles - Toutes les parties*, 2018.

[6] CEI 80848, *Langage de spécification GRAFCET pour diagrammes fonctionnels en séquence*, 1999.

[7] R. David et H. Alla, Du grafcet aux réseaux de Petri, Hermès - Lavoisier, 1992.

[8] CEI 62026-2, *Appareillage à basse tension – Interfaces appareil de commande-appareil (CDI) - Partie 2: AS-Interface*, 2008.

[9] CEI 61158, *Réseaux de communication industriels – Spécifications des bus de terrain - Toutes les parties*, 2014.

[10] CEI 61784-1, *Réseaux de communication industriels - Profils - Partie 1: Profils de bus de terrain*, 2014.

[11] ISO 11894, *Véhicules routiers — Gestionnaire de réseau de communication (CAN) - Toutes les parties*, 2004.

[12] Modbus Organisation, «http://www.modbus.org,» [En ligne]. [Accès le 2018].

[13] CEI 61784-2, *Réseaux de communication industriels - Profils - Partie 2 : Profils de bus de terrain supplémentaires pour les réseaux en temps réel basés sur l'ISO/CEI 8802-3*, 2014.

[14] CEI 62541, *OPC Unified Architecture – Toutes les parties*, 2014.

[15] CEI 60870, Matériels et systèmes de téléconduite - Toutes les parties, 2006.

[16] CEI 61850, *RÈseaux et systËmes de communication - toutes les parties*, 2016.

[17] E. Byres, D. Huffman et N. Kube, «On shaky ground - A study of security vulnerabilities in control protocols.,» 2006.

[18] E. Knapp, Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems,, Waltham, : Syngress, 2011.

[19] N. Falliere, L. Murchu et E. Chien, «W32.stuxnet dossier,» 2011. [En ligne]. Available: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf. [Accès le septembre 2018].

[20] R. Langner, «To Kill a Centrifuge,» novembre 2013. [En ligne]. Available: https://www.langner.com/wp-content/uploads/2017/03/to-kill-a-centrifuge.pdf. [Accès le septembre 2018].

[21] Dragos, *CrashOverride: Analysis of the Threat to Electric Grid Operations*, 2017.

[22] US-CERT, *CrashOverride*, 2017.

[23] R. Khan, P. Maynard, K. McLaughlin et others, «Threat Analysis of BlackEnergy Malware for Synchrophasor Based Real-time Control and Monitoring in Smart Grid,» chez *4th International Symposium for ICS & SCADA Cyber Security Research*, Swindon, 2016.

[24] H. Débar, M. Dacier et A. Wespi, «A revised taxonomy for intrusion-detection systems,» vol. 55, n° %17-8, pp. 361-378, 2000.

[25] C. Robert T. Marsh, «Critical Foundations: Protecting America's Infrastructures,» 1997.

[26] J. Larsen, *Breakage - Black Hat*, 2008.

[27] P. J. Ramadge et W. M. Wonham, «Supervisory Control of a Class of Discrete Event Processes,» *SIAM Journal on Control and Optimization (SICON)*, vol. 25, n° %11, pp. 206-230, 1987.

[28] M. Leucker et C. Schallhart, «A brief account of runtime verification,» *Journal of Logic and Algebraic Programming*, vol. 78, pp. 293-303, 2009.

[29] A. Pnueli, «The Temporal Logic of Programs,» chez *Proc. SFCS'77*, Washington, 1977.

[30] M. Leucker, «Runtime Verification for Linear-Time Temporal Logic,» chez *Engineering Trustworthy Software Systems: Second International School, SETSS 2016, Chongqing, China, March 28 - April 2, 2016, Tutorial Lectures*, J. P. Bowen, Z. Liu et Z. Zhang, Éds., Cham, Springer International Publishing, 2017, pp. 151-194.

[31] J. Ostroff et W. Wonham, «A framework for real-time discrete event control,» *IEEE Transactions on Automatic Control* , vol. 35, n° %14, pp. 386 - 397, 1990.

[32] D. O. Paun et M. Chechik, «On Closure Under Stuttering,» *FAC*, vol. 14, pp. 342-368, 2003.

[33] M. B. Dwyer, G. S. Avrunin et J. C. Corbett, «Patterns in property specifications for finite-state verification,» chez *Proc. ICSE'99*, 1999.

[34] O. Koucham, S. Mocanu, G. Hiet, J.-M. Thiriet et M. Frédéric, «Efficient Mining of Temporal Safety Properties for Intrusion Detection in Industrial Control Systems,» chez *Safeprocess* , Warsaw, Poland, 2018.

[35] C. Lemieux, D. Park et I. Beschastnikh, «General LTL Specification Mining,» chez *Proc. ASE'15*, 2015.

[36] W. L. W. Li, A. Forin et S. S. a. Seshia, «Scalable specification mining for verification and diagnosis,» chez *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, 2010.

[37] J. Yang, D. Evans, D. Bhardwaj et others, «Perracotta: Mining Temporal API Rules from Imperfect Traces,» chez *Proc. ICSE '06*, New York, NY, USA, 2006.

[38] M. Y. Vardi, «An automata-theoretic approach to linear temporal logic,» *Banff Higher Order Workshop 1995*, 1996.

[39] M. D'Amorim et G. Roşu, «Efficient Monitoring of ω-languages,» chez *Proc. CAV'05*, 2005.

[40] K. Havelund et G. Rosu, «Monitoring Programs Using Rewriting,» chez *Proceedings of the 16th IEEE International Conference on Automated Software Engineering*, Washington, 2001.

[41] K. Havelund et G. Rosu, «Testing Linear Temporal Logic Formulae on Finite Execution Traces,» RIACS, 2001.

[42] D. Basin, B. N. Bhatt et D. Traytel, «Almost Event-Rate Independent Monitoring of Metric Temporal Logic,» chez *Tools and Algorithms for the Construction and Analysis of Systems*, Berlin, 2017.

[43] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Nai Fovino et a. Trombetta, «A multidimensional critical state analysis for detecting intrusions in SCADA systems,» *IEEE Transactions on Industrial Informatics*, vol. 7, pp. 179-186, 2011.

[44] A. Carcano, I. N. Fovino, M. Masera et A. Trombetta, «State-based network intrusion detection systems for SCADA protocols: A proof of concept,» *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6027 LNCS, pp. 138-150, 2010.

[45] M. Caselli, E. Zambon et F. Kargl, «Sequence-aware Intrusion Detection in Industrial Control Systems,» chez *Proc. 1st ACM Workshop CPSS*, 2015.

[46] M. Yoon et G. Ciocarlie, «Communication Pattern Monitoring : Improving the Utility of Anomaly Detection for Industrial Control Systems,» chez *SENT*, 2014.

[47] Ö. Yüksel, J. Hartog et S. Etalle, «Reading Between the Fields: Practical, Effective Intrusion Detection for Industrial Control Systems,» chez *Proc. of the 31st Annual ACM Symp. on Applied Computing*, NY, 2016.

[48] N. Goldenberg et A. Wool, «Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems,» *International Journal of Critical Infrastructure Protection*, vol. 6, pp. 63-75, 2013.

[49] J. Schumann, P. Moosbrugger et K. Y. Rozier, «R2U2 : Monitoring and Diagnosis of Security Threats for Unmanned Aerial Systems,» chez *6th Int. Conf. Runtime Verification (RV'15), Vienna, Austria*, 2015.

[50] O. Koucham, S. Mocanu, G. Hiet et others, «Detecting Process-Aware Attacks in Sequential Control Systems,» chez *NordSec 2016*, Oulu, 2016.

[51] R. Alur, Henzinger et T.A., «A really temporal logic.,» chez *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, 1989.

[52] R. Koymans, «Specifying real-time properties with metric temporal logic,» *Real-time systems*, vol. 2, n° %14, pp. 255-299, 1990.

[53] S. Konrad et B. H. C. Cheng, «Real-time Specification Patterns,» chez *Proceedings of the 27th International Conference on Software Engineering*, New York, NY, USA, 2005.

[54] Y. Boussemart et M. L. Cummings, «Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models,» *Engineering Applications of Artificial Intelligence* , vol. 24, pp. 1252-1262, 2011.

[55] H. Debar et A. Wespi, «Aggregation and Correlation of Intrusion-Detection Alerts,» chez *RAID*, Berlin, 2001.

[56] A. Valdes et K. Skinner, «Probabilistic alert correlation,» chez *Int. Workshop on Recent Advances in Intrusion Detection*, 2001.

[57] F. Valeur, G. Vigna, C. Kruegel et others, «Comprehensive approach to intrusion detection alert correlation,» *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 146-169, 7 2004.

[58] O. Maler et D. Nickovic, «Monitoring temporal properties of continuous signals,» *FORMATS/FTRTFT*, p. 152–166, 2004.

[59] O. Maler et D. Nickovic, «Monitoring temporal properties of continuous signals.,» *International Journal on Software Tools for Technology Transfer*, vol. 15, n° %13, pp. 247-268, 2013.

[60] O. Maler, D. Nickovic et A. Pnueli, «Checking temporal properties of discrete, timed and continuous behaviors,» *Pillars of Computer Science*, p. 475–505, 2008.

[61] ANSI /IEEE Standard C37.2, *Electrical Power System Device Function Numbers, Acronyms, and Contact Designations*, ANSI/IEEE, 1996.

[62] NIST , *Smart Grid Framework 1.0*, 1999.

[63] CEI 61970-301, *Energy management system application program interface (EMS-API) - Common information model (CIM) base*, 2011.

[64] IEC 62351, *Power systems management and associated information exchange - Data and communications security*, 2007.

[65] CEI 62439, *Réseaux de communication industrielle : réseaux à haute disponibilité.*, 2013.

[66] ISO/IEC 9506, *Industrial Automation systems - Manufacturing Message Specification*, 2003.

[67] J. Hoyos, M. Dehus et T. Brown, «Exploiting GOOSE protocol: A practical attack on cyber-infrastructure,» chez *2012 IEEE Globecom Workshops*, 2012.

[68] M. Kabir-Querrec, S. Mocanu, J.-M. Thiriet et E. Savary, «Corrupted GOOSE Detectors: Anomaly Detection in Power Utility Real-Time Ethernet Communications,» chez *GreHack*, Grenoble, 2015.

[69] M. Kabir-Querrec, S. Mocanu, J.-M. Thiriet et E. Savary, «Cybersecurity of smart-grid control systems: Intrusion detection in IEC 61850 automation systems,» chez *RESSI*, Toulouse, 2016.

[70] M. Kabir-Querrec, S. Mocanu, P. Belmain, J.-M. Thiriet et E. Savary, «Architecture des systèmes d'automatisation des postes résiliente aux attaques des trames GOOSE,» chez *CAESAR*, Rennes, 2015.

[71] Merlin Gerin / Schneider Electric, *Protéction des réseaux électriques*, 2003.

[72] M. Kabir-Querrec, S. Mocanu, J.-M. Thiriet et E. Savary, «Power Utility Automation Cybersecurity: IEC 61850 Specification of an Intrusion Detection Function,» chez *25th European Safety and Reliability Conference (ESREL 2015)*, Zürich, 2015.

[73] M. Kabit-Querrec, «Cyber security of the smart grid control systems: intrusion detection in IEC 61850 communication networks,» Grenoble, 2017.

[74] RTE, *Contrôle commande des postes RTE - Modélisation IEC 61850*, 2018.

[75] M. Kabir-Querrec, S. Mocanu, J.-M. Thiriet et E. Savary, « A Test bed dedicated to the Study of Vulnerabilities in IEC 61850 Power Utility Automation Networks.,» chez *21st IEEE Emerging Technologies and Factory Automation*, Berlin, 2016.

[76] A. Altaher, *Implementation of a dependability framework for smart substation automation systems : application to electric energy distribution,* Thèse Grenoble, 2018.

[77] A. Altaher, J.-M. Thiriet et S. Mocanu, «Performance Evaluation of IEC 61850 safety-related communication architecture.,» chez *Summer School on Cyber-Physical Systems*, 2014, 2014.

[78] A. Altaher, S. Mocanu et J.-M. Thiriet, «Evaluation of Time-Critical Communications for IEC 61850-Substation Network Architecture.,» chez *International Conference Surveillance 8*, Roanne, 2015.

[79] A. Altaher, S. Mocanu et J.-M. Thiriet, «Dependability Optimization of Process-level Protection in an IEC-61850-Based Substation.,» chez *26th European Safety and Reliability Conference (ESREL 2016)*, Glasgow, 2016.

[80] A. Altaher, S. Mocanu et J.-M. Thiriet, «Experimental Evaluation of an IEC 61850-Station Bus Communication Reliability,» chez *JNCT 2015*, Valence, 2015.

[81] *Intrusion Detection Datasets DOI: 10.18709/perscido.2018.09.DS236,* 2018 .

[82] F. de Goër, C. Ferreira et L. Mounier, «SCAT: Learning from a single execution of a binary.,» chez *SANER*, Klagenfurt, 2017.

[83] F. de Goër, R. Groz et L. Mounier, «Lightweight heuristics to retrieve parameter associations from binaries.,» chez *PPREW@ACSAC Workshop*, Los Angeles, 2015.

[84] M. Shabaz et R. Groz, «Analysis and testing of black-box component-based systems by inferring partial models.,» *Software Testing, Verifification and Reliability*, vol. 24, n° %14, 2014.

[85] J. O. Kephart et D. M. Chess, «The vision of autonomic computing,» *IEEE Computer*, vol. 36, n° %11, pp. 41-50, 2003.

[86] J. Bai, S. Hariri et Y. Al-Nashif, «A Network Protection Framework for DNP3 over TCP/IP protocol,» chez *IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, 2015.

[87] B. Al Baalbaki, Y. Al-Nashif, S. Hariri et D. Kelly, «Autonomic Critical Infrastructure Protection (ACIP) system,» chez *AICCSA*, 2013.

[88] Q. Chen et S. Abdelwahe, « Towards realizing self-protecting SCADA systems,» chez *9th Annual Cyber and Information Security Research Conference (CISR '14)*, 2014.

[89] D. P. Cox, The Application of Autonomic Computing for the Protection of Industrial Control Systems, University of Arizona, 2011.

[90] Y. Lopes, N. C. Fernandes, D. C. Muchaluat-Saade et K. Obraczka, «ARES: An autonomic and resilient framework for smart grids,» chez *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Lisbonne, 2017.

[91] R. Fritz et P. Zhang, «Modeling and detection of cyber attacks on discrete event systems,» chez *14th IFAC Workshop on Discrete Event Systems WODES*, Sorrento, Italie, 2018.

[92] P. Lima et L. M. M. Carvalho, «Detectable and Undetectable Network Attack Security of Cyber-physical Systems,» chez *14th IFAC Workshop on Discrete Event Systems WODES*, Sorrento, Italie, 2018.

[93] L. Hélouët, H. Marchand et L. Ricker, «Opacity with powerful attackers,» chez *14th IFAC Workshop on Discrete Event Systems WODES*, Sorrento, Italie, 2018.

[94] C. Seatzu, «Partially observed discrete-event systems: from state estimation to intrusion detection,» chez *14th IFAC Workshop on Discrete Event Systems WODES*, Sorrento, Italie, 2018.

[95] E. Rutten, N. Marchand et D. Simon, «Feedback Control as MAPE-K Loop in Autonomic Computing,» chez *Software Engineering for Self-Adaptive Systems*, 2013.

[96] S. Mocanu et C. Commault, «Sparse representations of phase-type distributions,» *Communications in Statistics. Stochastic Models*, vol. 4, n° %1759-778, p. 15, 1999.

[97] C. Commault et S. Mocanu, «A generic property of phase-type representations,» *Journal of Applied Probability*, vol. 39, n° %14, pp. 775-785, 2002.

[98] C. Commault et S. Mocanu, « Phase-type distributions and representations: Some results and open problems for system theory,» *International Journal of Control*, vol. 76, n° %16, pp. 566 - 580, 2003.

[99] A. Royer, S. Mocanu et C. Commault, « Méthode de décomposition pour l'évaluation de performances de réseaux linéaires de routeurs ON/OFF,» *Revue des Sciences et Technologies de l'Information - Série TSI : Technique et Science Informatiques*, pp. 179-202, 2005.

[100] H. B. Nguyen, C. Commault et S. Mocanu, «Etude en simulation de politiques de pilotage multiproduit dans l'industrie des semi-conducteurs.,» chez *Les systèmes de production : applications interdisciplinaires et mutations*, 2007, pp. 33-44.

[101] S. Mihaita A. et S. Mocanu, «Simulation en temps continu pour la commande orientée événements des systèmes stochastiques à commutation,» *Journal Européen des Systèmes Automatisés*, vol. 45, n° %11-3, pp. 157-172, 2011.

[102] A. S. Mihaita et S. Mocanu, «An Energy Model for the Event-Based Control of a Switched Integrator,» chez *IFAC WC*, Milano, 2011.