République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique
Département d'Electronique
Laboratoire signal et communications

Thèse de Doctorat en Sciences

Spécialité Electronique

Développement d'un protocole SCADA temps réel sécurisé à base de l'IEC 60870-5-101

Présentée par :

Tarek CHERIFI

Sous la direction de :

Mme Latifa HAMAMI-MITICHE **Professeur** **ENP**

Membres de jury

Président :	**M. Chérif LARBES**	**Professeur**	**ENP**
Rapporteur :	**Mme Latifa HAMAMI**	**Professeur**	**ENP**
Examinateurs :	**Mme Saliha AOUAT**	**Professeur**	**USTHB**
	M. Rabah SADOUN	**Maître de conférences A**	**ENP**
	M. Rabah. GOURI	**Maître de conférences A**	**ENST**
	M. Omar NOUALI	**Directeur de recherche**	**CERIST**

ENP 2019

Ecole Nationale Polytechnique (ENP)
10, Avenue des Frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie
www.enp.edu.dz

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique
Département d'Electronique
Laboratoire signal et communications

Thèse de Doctorat en Sciences

Spécialité Electronique

Développement d'un protocole SCADA temps réel sécurisé à base de l'IEC 60870-5-101

Présentée par :

Tarek CHERIFI

Sous la direction de :

Mme Latifa HAMAMI-MITICHE **Professeur** **ENP**

Membres de jury

Président :	**M. Chérif LARBES**	**Professeur**	**ENP**
Rapporteur :	**Mme Latifa HAMAMI**	**Professeur**	**ENP**
Examinateurs :	**Mme Saliha AOUAT**	**Professeur**	**USTHB**
	M. Rabah SADOUN	**Maître de conférences A**	**ENP**
	M. Rabah. GOURI	**Maître de conférences A**	**ENST**
	M. Omar NOUALI	**Directeur de recherche**	**CERIST**

ENP 2019

Ecole Nationale Polytechnique (ENP)
10, Avenue des Frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie
www.enp.edu.dz

ملخص

تعد أنظمة المراقبة و التحكم عن بعد احد SCADA اهم الأنظمة المستخدمة لتسيير و استغلال البنى التحتية القاعدية، ونظرا لأهمية هذه الأنظمة فانه بات من الضروري تطوير أدوات حماية تتجاوب مع خصوصيات الأنظمة الصناعية بعدما اثبتت الجدران النارية التقليدية، آليات التوثيق، خوارزميات وبروتوكولات التشفير الحالية عجزها عن التوفيق بين كفاءة التشغيل و الحماية السبريانية المطلقة.

يقترح هذا المشروع منهجًا جديدًا لتأمين بروتوكول101-5-60870 IEC ، والذي يعد أحد اهم بروتوكولات SCADA المنفتحة، الغير موجهة والأكثر استخدامًا في مجال التحكم عن بعد. يدمج هذا الأسلوب طبقة الأمان بين الطبقة المادية وطبقة الوصلة لبنية الأداء المحسنة الخاصة بالبروتوكول. وتعتمد طبقة الأمان على تجسيد مفهوم الأمن الغير المشروط. تؤكد النتائج التجريبية أن منهجنا يحترم القيود الزمنية المفروضة من قب لأنظمة SCADA المستخدمة في شبكات توزيع الطاقة الكهربائية.

الكلمات الدالة

الحماية السيبرانية لأنظمة SCADA ,IEC 60870-5-101, الحماية اللامشروطة, الحماية المطلقة, الحماية المقواة , ST-101

Abstract

SCADA systems are used across the critical infrastructure to monitor and control vital industrial processes. Traditional firewalls, authentication mechanisms, and cryptographic algorithms and protocols are inadequate to secure SCADA systems and the underlying industrial processes from cyber attacks.

This work describes a novel approach for providing a high level of secrecy to the IEC60870-5-101protocol, an on-routable open SCADA communications protocol. The proposed approach incorporates a secrecy layer between the physical and link layers of the enhanced performance architecture of the IEC60870-5-101 protocol. The secrecy layer is an implementation of Shannon's notion of an unconditionally secure system in which perfect secrecy and strong ideal secrecy are leveraged to guarantee the authenticity, integrity and confidentiality of SCADA data transmission. Experimental results confirm that the proposed approach satisfies the temporal constraints imposed on SCADA systems used in electrical substations.

Keywords

SCADA security, IEC 60870-5-101, unconditional security, perfect security, ideal security, ST-101.

Résumé

Les systèmes SCADA sont utilisés par la visualisation et le télécontrôle des processus industriels vitaux de la majorité des infrastructures critiques. Les pare-feu traditionnels, les mécanismes d'authentification, les algorithmes et les protocoles cryptographiques actuels ne sont pas adaptés pour la sécurisation et la protection des systèmes SCADA face aux cyber-attaques.

Ce travail propose une nouvelle approche pour la sécurisation du protocole IEC 60870-5-101, un des protocoles de transmission SCADA ouverts et non-routables les plus utilisés en télécontrôle. Cette approche intègre une couche de sécurité entre la couche physique et la couche liaison de l'architecture à performance améliorée de l'IEC 60870-5-101. La couche de sécurité est une implémentation de la notion de sécurité inconditionnelle de Shannon. Des résultats expérimentaux confirment que notre approche satisfait les contraintes temporelles imposées par les systèmes SCADA utilisés dans les réseaux de distribution de l'énergie électrique.

Mots clés

Cyber sécurité des systèmes SCADA, IEC-60870-5-101, sécurité inconditionnelle, sécurité parfaite, sécurité idéale forte, ST-101

REMERCIEMENTS

Mes plus intenses remerciements vont, en premier lieu, à DIEU *le Miséricordieux le Tout-Puissant pour m'avoir donné le courage de mener à terme ce travail.*

Je tiens, en deuxième lieu, à remercier ma directrice de thèse, Madame HAMAMI*, pour son aide, son soutien et sa patience tout au long de mon travail de thèse de doctorat.*

Je tiens également à remercier Monsieur LARBES *d'avoir accepté de présider le jury auprès duquel seront exposés les résultats du travail effectué.*

Merci également aux membres examinateurs à savoir madame S.AOUAT, Professeure à l'USTHB, M.O. NOUALI Directeur de Recherche au CERIST, M.R. GOURI MCA à l'ENST et M. R. SADOUN MCA à l'ENP, d'avoir accepté de juger ce modeste travail.

Enfin, je suis reconnaissant à toutes celles et tous ceux qui ont contribué de près ou de loin à l'accomplissement de ce travail.

Table des matières

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ALGORITHMES

LISTE DES ABREVIATIONS

Abriviation	Description
AES	Advanced Encryption Standard
AIE	Application Information Elements
ANSI	American National Standards Institute
APCI	Application Protocol Control Information
ASCII	American Standard Code for Information Interchange
ASDU	Application Service Data Units
CIM	Computer Integrated Manufacturing
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
CS-PRNG	Cryptographically-Secure Pseudo-Random Number Generator
DCS	Distributed Control System
DDoS	Distributed Denial of Service Attack
DES	Data Encryption Standard
DNP3	Distributed Network Protocol version 3
DoS	Denial of Service attack
DUI	Data Unit Identification
EPA	Enhanced Performance Architecture
ERP	Enterprise Resource Planning
FAA	Federal Aviation Administration
FT-1.2	Frame Type 1.2
FT-3	Frame Type 3
GPS	Global Positioning System
IACS	Industrial Automation and Control Systems Security
ICS	Industrial Control System
IEC	International Electrotechnical Commission
IEC-101	IEC 60870-5-101
IED	Intelligent Electronic Device
IHM	Interface Homme Machine
IoT	Internet of Things
IP	Internet Protocol
ISA	International Society of Automation
ISO	International Standard Organization
IT	Information Technology
ITU-T	International Telecommunications Union - Telecommunications
LAN	Local Area Network
LPDU	Link Protocol Data Unit

LRC	Longitudinal Redundancy Check
MES	Manufacturing Execution System
MODBUS	MODicon communication BUS
MTU	Master Terminal Unit
NBS	National Bureau of Standards
NERC	North American Electric Reliability Council
NIST	National Institute of Standards & Technology
NSA	US National Security Agency
PLC	Programmable Logic Controller
PR-MIIT	PR-Ministry of Industry and Information Technology
PRNG	Pseudo-Random Number Generator
PR-SAC	PR-Standardization Administration of the People's Republic of China
RTOS	Real Time Opperating Systems
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SDoS	Simple Denial of Service Attack
SPP	Système Protection Profile
ST-101	Secure IEC 60870-5-101
STOE	System Target of Evaluation
T-101	IEC 60870-5-101
TC57	Technical Committee 57
TCP	Transmission Control Protocol
TNT	Dynamite trinitrotoluène
TPDU	Transport Protocol Data Unit
UDP	User Datagram Protocol
UK-CPNI	UK-Center for the Protection of National Infrastructure
US-DHS	US-Department of Homeland Security
US-DoE	US-Departement of Energy
US-GAO	US-Government Accountability Office
WAN	Wide Area Network

INTRODUCTION GENERALE

Les systèmes de supervision, contrôle et acquisition des données (SCADA) sont souvent utilisés pour conduire les processus industriels des infrastructures critiques tels que les réseaux électriques, les raffineries pétrochimiques, les pipelines transportant les hydrocarbures, les unités de traitement et de distribution des eaux et des eaux usées. Les systèmes SCADA sont conçus spécifiquement pour supporter le suivi et le contrôle, en temps réel et à distance, des installations industrielles distribuées sur une large zone géographique. Ces systèmes permettent aux opérateurs de prendre des décisions instantanées et précises pour assurer le bon fonctionnement des systèmes contrôlés. Ils offrent aussi des fonctionnalités indispensables telles que la supervision en temps réel, l'analyse instantanée des événements passés et courants, l'archivage des données et la génération des rapports, la simulation des situations et le partage d'information avec les autres systèmes de l'entité.

Ces importantes fonctionnalités font des infrastructures critiques actives fortement liées aux systèmes SCADA. Par conséquent, Chaque manipulation ou perturbation, accidentelle ou intentionnelle, peut avoir de sérieuses conséquences sociales, économiques, politiques ou environnementales. Avant l'an 2000, plus de 70% des incidents reportés affectant les systèmes SCADA étaient d'origines internes causées par des failles de conception des systèmes, des erreurs d'utilisation, des actes de vandalisme et de sabotage. A partir de l'an 2001, le nombre d'incidents dus aux cyber-attaques externes a remarquablement augmenté, neuf cyber-attaques enregistrés aux États-Unis en 2009, 39 en 2010, 198 en 2011. En réalité, les cyber-attaques externes sont estimées à plus de 70% des incidents à partir de 2012(Miller and Rowe 2012; Nicholson et al. 2012).

Les systèmes SCADA sont devenus l'une des cibles les plus attractives aux acteurs cybernétiques mal intentionnés. Les hackers, les hacktivistes, les amateurs, les criminels, les terroristes, et même les états et les nations hostiles sont intéressés à perturber ces systèmes critiques, maintenir un accès persistant à des fins malveillantes et voler la propriété intellectuelle et les données opérationnelles sensibles. Ces acteurs tirent parti des connaissances avancées sur les vulnérabilités des protocoles SCADA ouverts, des réseaux de télécommunications, des connectivités, des mécanismes de sécurité et des architectures matérielles et logicielles pour compromettre la disponibilité, l'intégrité et / ou la confidentialité des systèmes SCADA.

En raison du nombre croissant de vulnérabilités et de menaces pouvant affecter les infrastructures critiques, la sécurisation des systèmes SCADA est devenue une priorité absolue. Assurer l'authenticité et l'intégrité des messages SCADA est essentiel à la mise en œuvre de la sécurité du système. Par conséquent, il est essentiel qu'un système SCADA puisse vérifier la source et la destination de chaque message. En effet, si un dispositif externe, non autorisé ou compromis, devait être reconnu comme une entité système de confiance, le dispositif pourrait facilement être utilisé pour lancer des attaques passives, des attaques de modification et de fabrication et, dans certains cas, des attaques par déni de service. De même, avant qu'un opérateur puisse prendre une décision ou mettre en œuvre une action, le système SCADA doit fournir des informations complètes et correctes ; il est donc essentiel de s'assurer que tous les messages reçus sont complets et non modifiés.

En plus de l'authenticité et de l'intégrité, la confidentialité des messages est un aspect essentiel de la sécurité des systèmes SCADA. Leurs messages fournissent des informations en temps réel sur les infrastructures gérées, y compris leurs états, leurs activités et leurs faiblesses. De telles informations pourraient être exploitées par un acteur malveillant afin de déterminer le type d'attaques physiques et les moments idéaux pour un maximum d'impacts. Assurer la confidentialité protège également contre les attaques passives et préserve l'authenticité.

Des solutions informatiques sécuritaires traditionnelles sont souvent utilisées pour sécuriser les systèmes SCADA. Cependant, les pare-feu standards, les mécanismes d'authentification des utilisateurs et les algorithmes cryptographiques des technologies de l'information ne sont pas adaptés pour eux. Les systèmes SCADA ont des objectifs de confidentialité, des architectures sécuritaires, des technologies matérielles et logicielles et des exigences de qualité de service différentes. Plusieurs chercheurs ont présenté des approches pour analyser, détecter, mesurer et protéger les systèmes SCADA contre les cyber-attaques et atténuer les conséquences négatives de ces attaques (Cherifi and Hamami 2018).

Les auteurs des articles (Kriaa 2016) et (Huang et al. 2009) ont modélisé les cyber-attaques visant les systèmes cyber-physiques par des fonction de transfert symbolisant les attaques internes et les attaques externes passives, DoS et les attaques par modification. L'article (Nazir et al. 2017) a exposé l'ensemble des plates-formes, simulateurs, bancs de tests, et modèles mathématiques et probabilistes des attaques visant les systèmes SCADA. D'autre part, l'article (Knowles et al. 2015) a

présenté les principaux mécanismes de la gestion de la sécurité des systèmes de contrôle industriel. Une approche expérimentale pour l'évaluation de ces mécanismes a été présentée dans (Genge et al. 2015). L'étude de ces mécanismes et des différentes normes et stratégies nationales, régionales et internationales (Byres et al. 2012; Trappey et al. 2017; Zhou et al. 2017) nous a montré que la protection du segment de coopération des systèmes SCADA a été proposée sans prendre en considération les spécificités des réseaux industriels.

A l'exception du protocole SCADA de transport Modbus, rares sont les travaux qui se sont intéressés à la cyber-sécurité des protocoles de transport SCADA: dans l'article (Huitsing et al. 2008), les auteurs ont présenté une étude sur les taxonomies des cyber-attaques connues spécifiques au Modbus TCP/IP. Les articles (Erez and Wool 2015) et (Goldenberg and Wool 2013) se sont intéressés à la classification, la modélisation et la détection des intrusions visant les systèmes à base de Modbus.

Cependant, ce travail présente notre nouvelle approche développée pour sécuriser les communications SCADA contre les attaques passives, de fabrication et de modification externes. Il est le premier travail qui s'intéresse à fournir un niveau élevé de sécurité au protocole IEC-60870-5-101, le protocole de communication SCADA ouvert non routable utilisé dans l'industrie mondiale de l'énergie électrique.

L'approche proposée incorpore une couche Sécurité entre les couches physique et de liaison de l'architecture de performance améliorée du protocole IEC-60870-5-101. La couche Sécurité utilise des générateurs de nombres pseudo-aléatoires hautement sécurisés et des fonctions de hachage pour assurer la mise en œuvre en temps réel de la notion de système inconditionnel de Shannon, dans laquelle les principes de la confidentialité parfaite et confidentialité idéale forte sont utilisés.

Des résultats expérimentaux confirment que l'approche proposée répond aux contraintes temporelles imposées aux systèmes SCADA utilisés dans la gestion des postes électriques.

Dans ce contexte, nous avons divisé cette thèse en quatre chapitres. Le premier chapitre expose des généralités sur les architectures matérielles et protocolaires des systèmes SCADA. Dans le deuxième chapitre, nous nous sommes intéressés à l'aspect sécuritaire des systèmes SCADA. Il le compare à l'aspect sécuritaire des systèmes IT, Il introduit les principaux auteurs des cyber-attaques, leurs cibles et leurs outils, il illustre notre model développé pour analyser les cyber-attaques visant les

systèmes SCADA et finalement il dresse un état de l'art sur les principaux standards et stratégies de sécurité SCADA établis. L'analyse de ces derniers nous a montré que les vulnérabilités liées à la sécurisation des trames transport ont été négligées. Notre contribution est de proposer un nouveau crypto-système adapté aux protocoles de transport SCADA. Pour ce faire, le troisième chapitre trace les grandes lignes de notre crypto-système adressé à la sécurisation des trames de transport en bénéficiant des caractéristiques des crypto-systèmes IT les plus reconnus pour leurs niveaux de confidentialité élevés. Le dernier chapitre présente une description détaillée de notre protocole SCADA de transport T-101 que nous avons développé en sécurisant le protocole IEC 60870-5-101 avec notre crypto-système, ainsi qu'une analyse et une discussion exprimée à la base des problèmes rencontrés et des résultats expérimentaux obtenus.

CHAPITRE I : GENERALITES SUR LES SYSTEMES SCADA

I. Introduction

L'industrie de l'électricité, l'approvisionnement en eaux potables, le traitement des eaux usées, la gestion des réseaux de télécommunication, la canalisation des hydrocarbures, le transport ainsi que de nombreuses autres infrastructures critiques exploitent les systèmes de supervision, de contrôle et d'acquisition des données (SCADA – Supervisory Control And Data Acquisition) pour assurer une gestion fiable et une maintenance instantanées des installations. Un système SCADA est un système de contrôle industriel (ICS – Industrial Control System) temps-réel centralisé, conçu pour échanger des données de télévisualisation et de télécontrôle entre les installations techniques distribuées sur une large zone géographique d'une part et un centre de contrôle d'autre part. Il fournit toutes les informations nécessaires pour aider les opérateurs à prendre des décisions rapides et correctes afin de maintenir le bon fonctionnement des systèmes contrôlés en collectant des informations instantanées et en délivrant des instructions mieux adaptées. Il permet également des signalisations efficaces des perturbations, des localisations rapides des défauts, des interventions de maintenance optimale à distance, des réductions des risques, une minimisation des coûts d'exploitation et des prédictions précises basée sur des statistiques en temps réel.

En fonctionnement normal, le rôle principal des systèmes SCADA est de prendre des décisions instantanées, adaptées aux informations actualisées, conformément aux degrés de liberté exigés par la flexibilité décisionnelle des systèmes. Pour cela, ils sont amenés à faire de l'ordonnancement temps réel, de l'optimisation, à modifier en ligne la commande et à gérer le passage d'un algorithme de surveillance à un autre.

En présence d'une anomalie, la supervision doit prendre toutes les dispositions nécessaires pour le retour vers le fonctionnement normal par la détermination d'un nouveau mode de fonctionnement. Il s'agit de choisir une solution curative, d'effectuer des ré ordonnancements "locaux" ou de déclencher des procédures d'urgence, etc(Bailey and Wright 2003).

Afin de répondre à ses différentes fonctions, les systèmes SCADA doivent à la fois fonctionner en temps réel sur des équipements industriels, généralement, de faibles ressources avec une très haute fiabilité. Chaque perturbation peut avoir de graves conséquences sur les infrastructures critiques gérées.

Malgré leurs conceptions basées sur des technologies informatiques et de communications modernes connues, les systèmes SCADA ont adopté des architectures matérielles et logicielles spécifiques afin de répondre aux exigences fonctionnelles et temporelles imposées par le milieu industriel. Ce chapitre présente un aperçu général sur les architectures matérielles et logicielles (protocolaires) des systèmes SCADA modernes utilisés pour la téléconduite des réseaux.

II. Architecture matérielle des systèmes SCADA modernes

Du point de vue topologie, les systèmes SCADA sont des réseaux informatiques industriels centralisés composés de trois segments : **(1)** Le segment principal du réseau SCADA, **(2)** le segment des équipements de terrain et **(3)** le segment de réseau de coopération. Le segment principal du réseau SCADA, également appelé Unité Terminale Principale (MTU- Master Terminal Unit), est installé au niveau d'un centre de contrôle. Il regroupe l'ensemble des dispositifs et équipements nécessaires pour la réception, le traitement, la visualisation des informations sur les installations distantes surveillées d'une part et pour lancer des instructions de contrôle d'autre part. Le segment des équipements de terrain regroupe l'ensemble des dispositifs installés sur les structures techniques distantes telles que les Unités Terminales Distantes (RTU- Remote Terminal Unit), les dispositifs électroniques intelligents (IED- Intelligent Electronic Device), les Automates Programmables Industriels (PLC- Programmable Logic Controller), ainsi que les capteurs et les actionneurs. Le segment de réseau de coopération représente tous les réseaux de télécommunication explorés pour assurer la communication entre les équipements du segment principal du réseau SCADA et les équipements de terrain(Clarke et al. 2004; Igure et al. 2006; McDonald 2012).

En général, un réseau SCADA assure une commande centralisée par l'usage d'une seule MTU communiquant avec de nombreuses RTU. Chaque RTU assure une communication locale avec un ensemble d'autres équipements de terrain de type IED, PLC, capteurs et/ou actionneurs. Dans la direction de visualisation, la RTU recueille, organise puis envoie les données reçues d'autres équipements de terrain au MTU. Le MTU traduit, enregistre, présente, affiche et offre toutes les

données nécessaires aux opérateurs et aux superviseurs. Cependant, dans le sens du contrôle, le MTU envoie des instructions de demande et de contrôle aux RTU qui les analysent puis les retransmet aux autres équipements de terrain pour l'exécution des instructions ou réponse aux interrogations. La Figure I.1 représente l'architecture matérielle des systèmes SCADA.

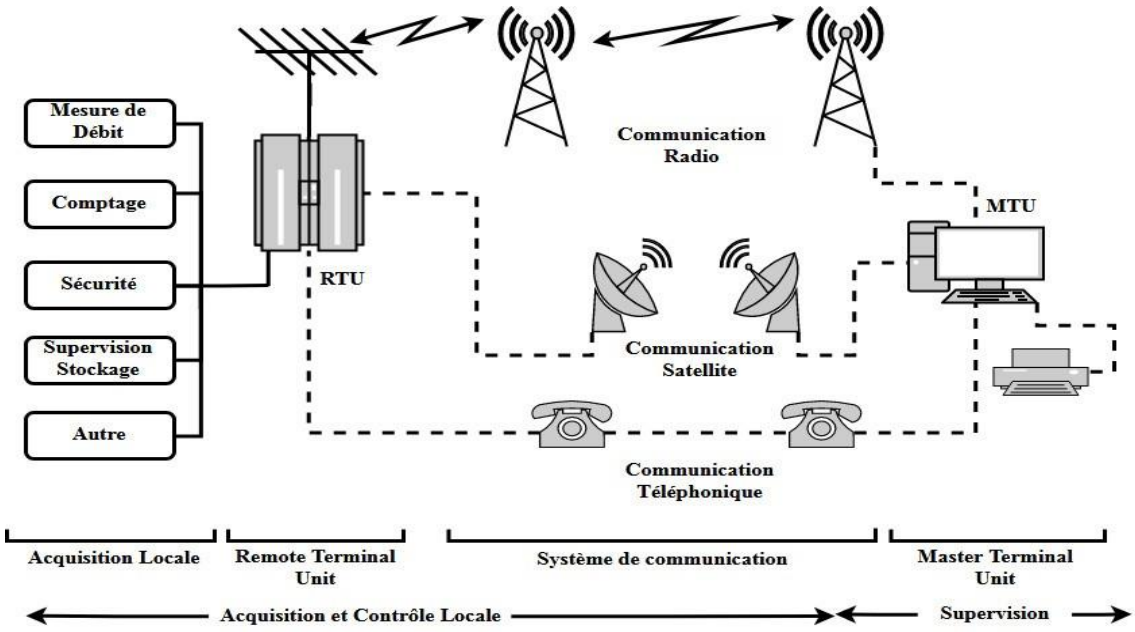


Figure I.1 Architecture matérielle des systèmes SCADA.

Le réseau de coopération représente la colonne vertébrale du système. Il se charge de la transmission de toutes les données de supervision et de contrôle entre les autres segments du système. Pour ceci, son état de fonctionnement est très critique pour la fiabilité du système. La technologie de communication choisie doit être crédible et optimale pour répondre aux exigences du système SCADA en respectant les caractéristiques physiques du canal, la bande passante disponible, le débit de communication offert, le type de multiplexage et de méthodes d'accès, le mode de communication défini (simplexe, half-duplex ou full-duplex), la taille du réseau, la mobilité du système, le niveau de la sécurité et surtout de son coût et de la réglementation imposée.

III. Évolution de l'architecture des systèmes SCADA

Afin d'améliorer sa performance, l'architecture des systèmes SCADA modernes n'a pas cessé de se réformer en bénéficiant de l'évolution des technologies de télécommunication et de développement des réseaux d'information (Nicholson et al. 2012; Stouffer et al. 2006; Stouffer et al. 2014).

La première génération d'architecture SCADA contient les systèmes SCADA monolithiques (Stouffer et al. 2006). En effet, lors de l'apparition des premières versions SCADA, le concept des réseaux n'était pas encore développé. Le partage des connexions et la collaboration entre les différents systèmes étaient très limités voir absents. Les MTU étaient des systèmes primaires basés sur des machines mainframes. Elles utilisaient des protocoles de communication SCADA propriétaires pour communiquer avec un nombre très limité de RTU. L'échange de données n'était possible qu'à l'intérieur du même système à travers des réseaux étendus (WAN - Wide Area Network) conçus spécialement pour être utilisés pour les communications SCADA sans aucun partage avec d'autres systèmes.

Du point de vue logiciel, les applications SCADA de cette génération étaient basées sur une architecture 1-Tier où les couches Interface Homme - Machine (IHM), la couche de traitement et la couche de gestion des données sont implémentées sur la même interface. L'application utilisait les disques de la machine pour traiter les données et pour stocker les résultats.

La deuxième génération d'architecture SCADA est connue par les systèmes SCADA distribués(Stouffer et al. 2006). Leurs MTU contiennent un ensemble de mini-ordinateurs interconnectés par réseaux locaux (LAN) qui assurent une communication et un traitement des données en temps réel. Cette architecture augmente la puissance de traitement et améliore la redondance et la fiabilité du système. Cependant, les systèmes distribués continuent d'utiliser des protocoles propriétaires et des liaisons WAN spécialisées.

La deuxième génération a séparé la couche interface homme machine des autres couches pour faire face à l'augmentation du volume des données. Elle est devenue basée sur une architecture 2-Tier, ou encore client / serveur.

La génération des systèmes SCADA actuels est basée sur une architecture en réseaux ouverts(Stouffer et al. 2006). Les systèmes SCADA utilisent des protocoles SCADA ouverts et des technologies WAN communes. Cette architecture a permis aux systèmes d'être (1) plus développés, (2) plus indépendants des fournisseurs, (3) plus riche en fonctionnalités, (4) moins chers, (5) plus simples,(6) plus rapides à déployer, (7) plus fiables, (8) plus connectés et (9) plus interactifs avec les autres systèmes d'entreprise. Les équipements des MTU peuvent être déployés sur des réseaux WAN et non pas uniquement sur des réseaux LAN. La Figure I.2 représente un exemple d'une architecture actuelle d'une MTU moderne.

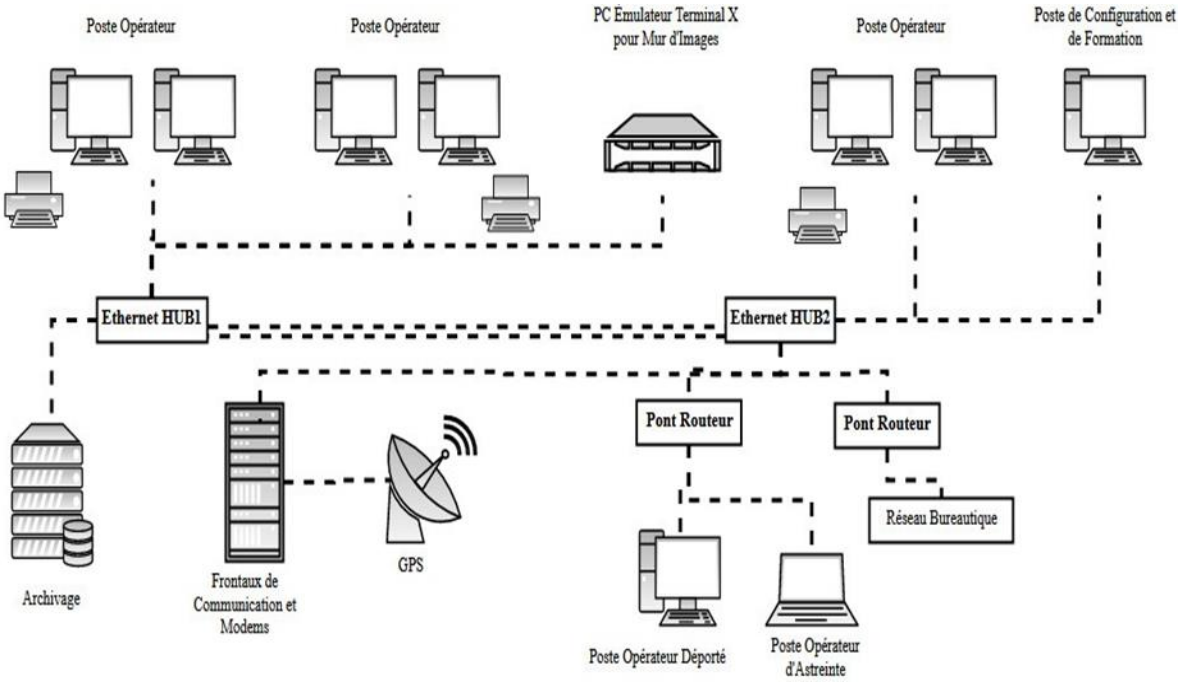


Figure I.2 Architecture d'une MTU moderne.

Avec le développement des technologies WEB, la troisième génération a décomposé la couche applicative en s'étalant sur une architecture 3-Tier composée du client, demandeur de ressources, du serveur d'application (middleware), chargé du traitement des données, et du serveur de base de données, fournisseur des données au serveur d'application.

La prochaine génération d'architecture SCADA est en cours de développement. Plusieurs travaux ont mentionné qu'elle peut être définie comme une architecture SCADA basée sur le concept du Cloud-computing, où les nœuds du système, principalement les MTU, sont implémentés sur des machines et des réseaux virtuels du Cloud (Battistelli et al. 2018; Bui et al. 2016; Cherifi and Hamami 2018; Saravanan et al. 2018). Cette architecture est plus adaptée aux nouveaux équipements SCADA utilisant des protocoles de transmission SCADA routables et des protocoles de l'internet des objets (IoT - Internet of Things). Elle réduit considérablement les coûts d'infrastructure et augmente la facilité d'entretien et d'intégration. En conséquence les systèmes SCADA peuvent désormais donner l'état en temps réel et utiliser les facteurs d'échelle permis par le cloud-computing pour mettre en œuvre des algorithmes de contrôle plus complexes que ceux supportés par les systèmes actuels.

Une architecture N-Tier, qui introduit des couches de virtualisation et de communication sur le cloud, semble être l'architecture la mieux adaptée pour la quatrième génération.

IV. Architecture protocolaire des systèmes SCADA modernes

Les protocoles SCADA de communication représentent les langages et les manières utilisés par les différents composants du système pour échanger leurs données. Ces protocoles sont conçus spécialement pour assurer un fonctionnement temps réel des infrastructures critiques en s'adaptant aussi bien aux technologies de communication mises à disposition qu'aux besoins spécifiques d'exploitation.

Du point de vue architecture protocolaire, les protocoles SCADA sont classifiés comme protocoles applicatifs qui appartiennent à la couche d'application du modèle TCP/IP. Afin d'augmenter les performances des protocoles, l'architecture à performances améliorées (EPA – Enhanced Performance Architecture) a été développée en se basant sur une architecture protocolaire en couches(Clarke et al. 2004). Elle regroupe les couches Accès au réseau, Internet et Transport du modèle TCP/IP en une seule couche dite Physique ; et elle divise la couche Application du modèle TCP/IP en deux couches principales (la couche Liaison de données et la couche Application) et deux Pseudo-couches facultatives (La pseudo-couche Pseudo-Réseau et la Pseudo-Couche Pseudo-transport).

La couche Liaison du modèle EPA se charge de l'identification des équipements communicants, des fonctions ciblées et de l'intégrité des messages échangés. La Pseudo-couche Pseudo-réseau définit le type du routage logique idéal pour acheminer les paquets envoyés alors que la Pseudo-couche Transport propose principalement les méthodes de segmentation de longues données. Finalement, la couche Application définit les caractéristiques des objets manipulés pour la télégestion des infrastructures. Une couche supplémentaire, dite couche Processus de l'utilisateur, peut s'ajouter à l'architecture pour définir les spécificités des options d'interopérabilité choisies lors de l'implémentation par rapport aux définitions globales des protocoles. La Figure I.3 représente la correspondance entre les couches des trois modèles en couches : ISO, TCP/IP et EPA.

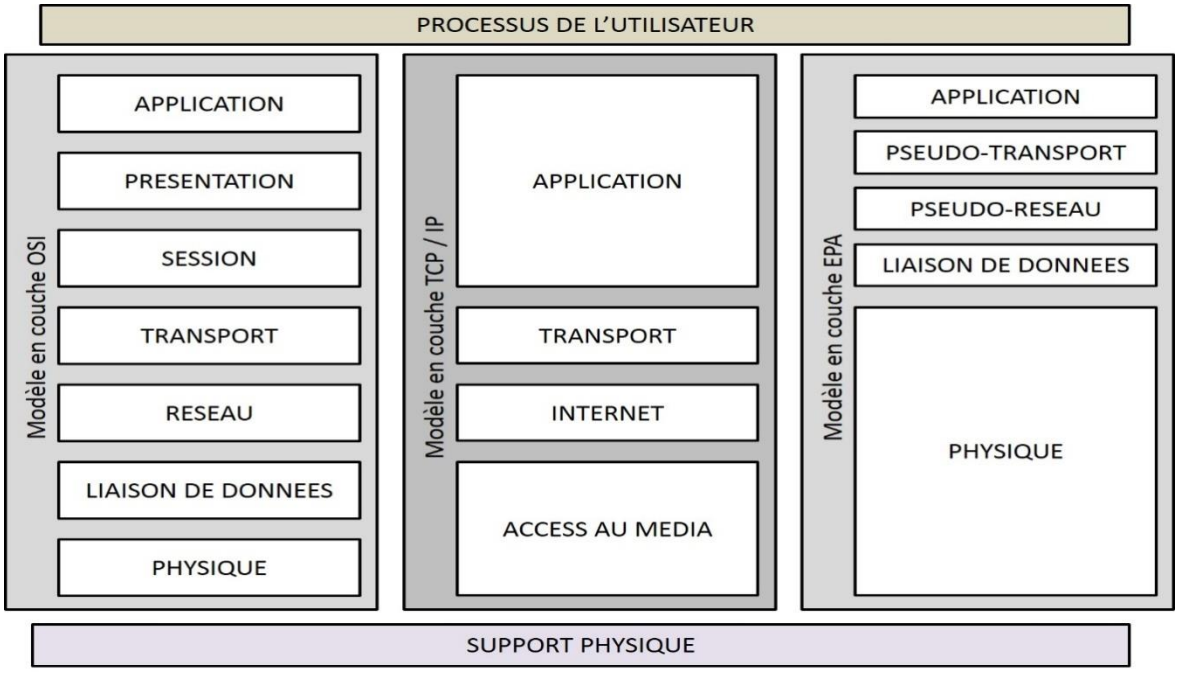


Figure I.3 Représentation des modèles en couches ISO, TCP/IP et l'EPA

En se basant sur l'architecture protocolaire en couches des systèmes SCADA, plusieurs critères de classification ont été définis dont nous citons :

1. Classification selon la topologie physique

Les trois topologies physiques les plus utilisées pour les systèmes SCADA sont (1) la topologie point à point, (1) la topologie multipoint et (3) la topologie station relais(Clarke et al. 2004). La configuration en liaison point à point est la configuration la plus simple à employer. Elle est formée de couples de ports de communication, chaque liaison est représentée par un couple unique dédié qui permet d'établir des communications en mode full-duplex en utilisant deux fréquences séparées ou en half-duplex en utilisant la même fréquence.

La configuration physique point – multipoint se traduit par le pouvoir d'un port d'une station de lancer des communications avec différents autres ports simultanément. L'implémentation de cette topologie nécessite l'utilisation d'algorithmes de gestion de trafics pour éviter les collisions entre les messages des différentes stations voulant transmettre en même temps.

La topologie de la station relais est utilisée soit pour le cas d'absence de visibilité directe entre les stations ou pour le cas de non-conformité du protocole entre la MTU et les RTU. Une RTU intermédiaire est configurée comme une station relais chargée de renvoyer les données interceptées

en amplifiant le signal de communication et en utilisant des protocoles supportés par l'ensemble des stations réceptrices.

2. Classification selon l'architecture réseau

En se basant sur le modèle TCP /IP, les protocoles SCADA peuvent être divisés en protocoles routables et protocoles non-routables. La différenciation entre les réseaux routables et non routables est de moins en moins fréquente à mesure que les communications industrielles se déploient de plus en plus sur réseau IP(Gao et al. 2014; Knapp and Langill 2014).

Un réseau non-routable fait référence aux liaisons de communication série, bus et point à point qui utilisent les réseaux du contrôle industriel. Un réseau routable signifie généralement un réseau utilisant le protocole Internet IP bien que d'autres protocoles routables, tels que AppleTalk, DECnet, Novell IPX et d'autres protocoles de réseau hérités s'appliquent certainement. Les réseaux routables incluent également des variantes routables des premiers protocoles ICS non routables qui ont été modifiés pour fonctionner sur TCP / IP, tels que Modbus sur TCP / IP, Modbus / TCP et DNP3 sur TCP / UDP.(Knapp and Langill 2014)

3. Classification selon le mode de communication

Deux modes de communication sont utilisées lors des échanges des données entre les différents composants du système SCADA : le mode de communication symétrique (balanced communication) et le mode de communication asymétrique (Unbalanced communication)(Bailey and Wright 2003; Clarke et al. 2004).

Le mode de communication symétrique, connu sous le nom Maître/ Maître (Master/Master), ne présente aucune gestion principale de la communication, les stations sont indépendantes et doivent se concurrencer pour avoir l'accès au media de transmission, ce qui rend les collisions inévitables. Ce mode est favorable pour assurer la communication d'un grand nombre de postes d'importance normale qui ne nécessitent pas une interrogation continue, dans ce cas le protocole SCADA doit être implémenté sur des liaisons point à point ou accompagné d'une méthode de gestion de collision.

Dans le mode asymétrique, connu par Maître / Esclave (Master / Slave), le maître, (MTU), contrôle totalement le système de communication. Uniquement lui, peut initier une communication en effectuant des interrogations répétitives à ses esclaves (RTU) afin de détecter toutes défaillances qui peuvent les affecter ou / et toute nouvelle information sur le système contrôlé.

Un maître peut communiquer avec plusieurs esclaves sans collision en précisant, dans chaque trame émise l'adresse de l'esclave interrogé. Du côté esclave, la communication est uniquement événementielle. En effet, il ne peut répondre que sur les interrogations en indiquant les changements d'état survenus sur une de ses variables.

Contrairement à la communication symétrique, la communication asymétrique est une méthode de gestion de collisions. Elle peut être implémentée directement sur n'importe quelle topologie ; elle permet la détection des défaillances des stations en temps réel (une station qui ne répond pas est une station défaillante) ainsi que la définition et la gestion des priorités des stations esclaves en modifiant leurs fréquences d'interrogation (une station prioritaire est une station plus interrogée).

Le seul inconvénient de ce mode de communication s'illustre en présence d'un nombre important de stations esclaves à gérer où le temps entre deux interrogations devient significatif. Dans ce cas, les concepteurs de solutions SCADA font recourt à la répartition des esclaves en groupe de communication en augmentant le nombre de ports utilisés ou en utilisant des concentrateurs d'information (architecture de la station relais).

4. Classification selon la pyramide CIM

En se basant sur la norme ANSI/ISA-95 publié aussi sous la référence ISO/CEI 62264, l'industrie moderne assistée par ordinateur (CIM - Computer Integrated Manufacturing) peut s'organiser sous forme d'une pyramide en couches fonctionnelles. Cette pyramide est divisée en deux parties : La première partie est dite partie de *Supervision & Contrôle* qui se charge de l'acquisition des données industrielles et de l'exécution des consignes d'exploitation. La deuxième partie est dite partie *Prise de décision* qui présente tous les outils nécessaires pour l'analyse les données d'exploitation et pour la définition des consignes de gestion.

La partie Supervision & Contrôle est composée de trois couches fonctionnelles : (1) La couche de niveau 0 dite *couche Opérative* qui comporte les équipements de base tels les capteurs et les actionneurs. (2) La couche de niveau 1 est la couche *Automatisme* composée des équipements dotés d'un certain niveau d'intelligence qui permet d'automatiser les processus tels les PLC et les IDE. (3) La dernière couche de cette partie est la couche de niveau 2, dite couche de *Supervision*, qui comporte l'ensemble des systèmes de contrôle industriel ICS dont nous trouvons les systèmes SCADA, DCS et les Smart-Grids.

La partie Prise de décision est composée de deux couches : (1) La première couche est la couche de niveau 3 ou la couche de *Gestion des processus industriels* (MES - Manufacturing Execution System) qui se base sur un système informatique permettant l'optimisation des processus de production en analysant des informations instantanées sur le système productif. (2) La deuxième couche est la couche de niveau 4 dite couche *Progiciel de gestion intégré* (ERP- enterprise resource planning) qui représente une solution complète pour la définition, le suivi et la gestion des plans de développement (business plan) des entreprises industrielles.

La Figure I.4 représente la structure en couches de la pyramide CIM ainsi que les catégories de protocoles utilisés dans les systèmes SCADA.

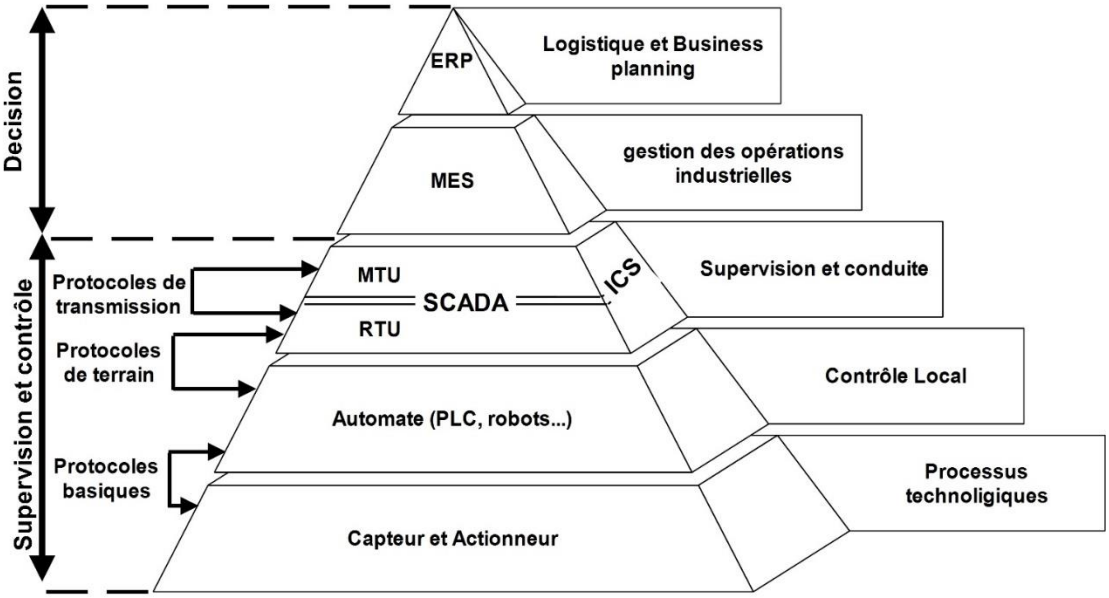


Figure I.4 Pyramide CIM

En se basant sur la pyramide CIM, nous constatons que nous pouvons classifier les protocoles de communication SCADA en (1) protocoles basiques, (2) protocoles de terrain et (3) protocoles de transport (nommés aussi protocoles de transmission).

Les protocoles basiques sont des protocoles de communication du niveau le plus bas qui assurent la communication entre les équipements opératifs et les automates à l'aide de trames de format très simple et de taille de données très réduite. Les protocoles de terrain sont des protocoles de proximité qui assurent une communication locale numérique entre les automates et les RTU avec des trames ayant une taille de données plus importante tels que l'IEC-60870-5-103, l'IEC-61850, Profibus, CAN, Modbus, 1-ware... Finalement, les protocoles de transport sont des protocoles bien structurés

qui se chargent de transporter des flux importants de données entre les MTU et les RTU sur des réseaux larges WAN. Nous citons parmi ces protocoles l'IEC 60870-5-101/104, le DNP3, ou certaines variantes de Modbus.

Nous nous intéressons, dans ce qui suit, à la présentation des principaux protocoles de transport utilisés dans la téléconduite des réseaux.

V. Principaux protocoles de transports utilisés dans la téléconduite

Les protocoles de transport SCADA sont des protocoles temps réel conçus pour assurer une transmission fiable d'un flux important de données entre les MTU et les RTU. Dans le domaine de la téléconduite, nous distinguons trois familles de protocoles de transport les plus utilisés: (1) le IEC 60870-5, (2) le DNP3 et (3) MODBUS.

1. Le protocole IEC 60870-5-101

Le protocole de transport de données SCADA IEC 60870-5-101, connu par IEC-101 ou encore T-101, est l'un des protocoles SCADA ouverts non-routables des plus répandus. Appartenant à la famille des protocoles IEC 60870-5, il est, essentiellement, adopté pour assurer une communication série fiable entre les MTU et les RTU des systèmes de téléconduite des différentes infrastructures critiques en Europe, Afrique du nord, Moyen-Orient, Chine et dans de nombreuses autres régions du monde.

Le Comité Technique d'études 57 (TC-57 - Technical Committee 57) de la Commission Électrotechnique Internationale (IEC - International Electrotechnical Commission) l'a conçu comme un protocole temps réel capable de transmettre les données de téléconduite des équipements et des installations techniques électriques dispersés sur de larges zones géographique. Il assure l'interopérabilité entre les MTU et les RTU des systèmes SCADA. Défini comme la première norme d'accompagnement ouverte, il est dérivé des cinq documents normatifs spécifiant les références des protocoles de transport de données SCADA des systèmes de téléconduite IEC60870-5(Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014): (1) le document *IEC60870-5-1* (1990) intitulé « *Formats de trames de transmission »* ; (2) le document *IEC60870-5-2* (1992) intitulé « *Procédures de transmission de liaison de données* »;(3) le document *IEC60870-5-3* (1992) intitulé « *Structure générale des données d'application* »; (4) le document *IEC60870-5-4*: (1993) intitulé « *Définition et*

codage des éléments d'information d'application »; et finalement (5) le document *IEC60870-5-5*: (1995) intitulé « *Fonctions d'application fondamentales* ».

L'architecture protocolaire de l'IEC60870-5-101 est basée sur une architecture à performances améliorées (EPA - Enhanced Performance Architecture) à trois couches. La première couche est la couche physique qui définit les interfaces séries utilisées en se basant sur les normes ITU-T. La deuxième couche représente la couche Liaison de données qui détermine les procédures et les modes de transmission, ainsi que les formats de trames utilisés tels qu'ils sont décrits par les références sélectionnées de l'IEC60870-5-1 et de l'IEC60870-5-2. La couche application définit les unités de données de service d'application (ASDU - Application Service Data Units) appropriées et les spécifications de codage des éléments d'information d'application (AIE - Application Information Elements) selon les références sélectionnées de l'IEC 60870-5-3 et l'IEC 60870-5-4. Le processus de l'utilisateur est une couche supplémentaire ajoutée pour définir les processus et les fonctions d'application spécifiques à la téléconduite selon les références sélectionnées de l'IEC 60870-5-5 (Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014).

La Figure I.5 illustre la relation entre l'architecture EPA de l'IEC-101 et les documents IEC-60870-5.

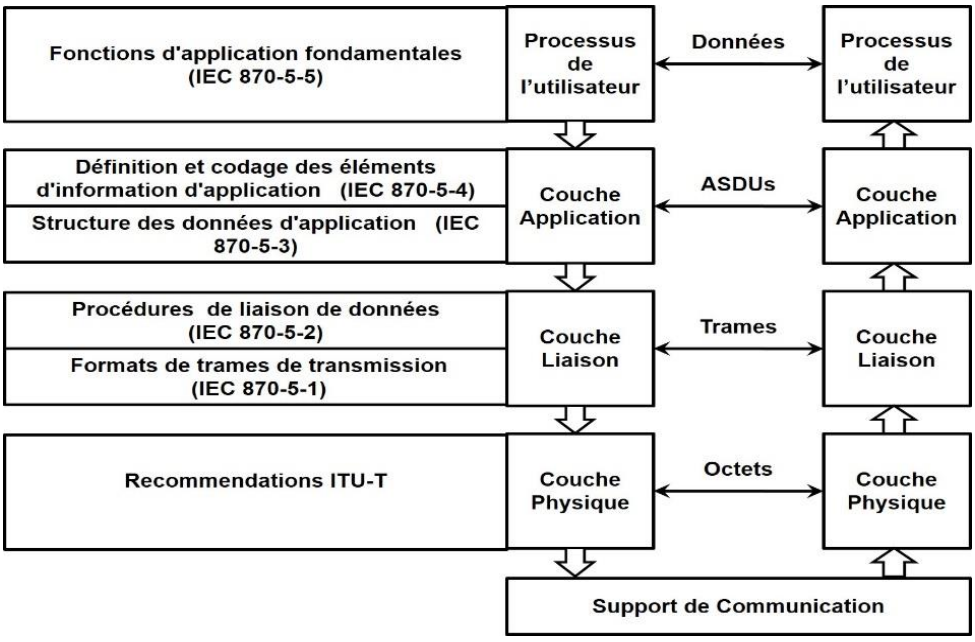


Figure I.5 Relation entre l'architecture EPA de l'IEC-101 et les documents IEC-60870-5.

La couche physique de l'IEC-101 est responsable de l'émission et la réception des bits et des octets à travers le support physique. Elle propose l'utilisation des interfaces de communication asynchrones ITU-T V.24/ ITU-T V28 avec un débit maximal de 9600 bits/s, ou des interfaces synchrones X.24 / X.27 avec un débit maximal de 64000 bits/s. Il est important de noter que le protocole autorise l'utilisation d'autres interfaces de communication séries. La couche physique de l'IEC-101 lui permet de supporter les configurations de réseaux point à point, point à point multiple, multipoint en étoile, multipoint en ligne partagée et multipoint en anneau(Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014).

La couche liaison n'adopte aucune méthode d'accès multiple au canal ou de gestion de collisions. Cela ne l'empêche pas d'établir des communications avec les deux modes de transmission (symétrique et asymétrique). Le mode de communication symétrique « le mode maître/maître » ne peut être établi que sur des infrastructures réseaux point à point. Par contre, le mode asymétrique « le mode maître/esclave » est adapté à l'ensemble des configurations acceptées en considérant, toujours, les RTU comme esclaves(Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014).

D'autre part, la couche liaison de données définit trois procédures de transmission standards. (1) La procédure *Envoi / Pas de réponse* est utilisée pour l'envoi des trames de diffusion et des trames qui n'exigent pas de réponses. (2) La procédure *Envoi/Confirmation* est utilisée pour l'émission des trames importantes qui nécessitent une confirmation de la réception des trames. Finalement, (3) la procédure *Demande/ Réponse* est utilisée pour envoyer les trames qui interrogent les RTU sur l'état des installations et qui nécessitent une transmission d'information d'exploitations(Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014).

Du point de vue structure de trames, la couche liaison définit le format et les procédures utilisés pour envoyer la totalité des trames sans erreurs. Elle utilise le format de trames *FT-1.2* défini par l'IEC-60870-5-1. Le format FT-1.2 présente trois formes de trames: (1) Les trames *IEC-101 de longueur variable* utilisées pour la transmission des données d'utilisateurs ;(2) les trames *IEC-101 de longueur fixe* utilisées pour les trames qui ne portent pas des données d'utilisateur ; et (3) les *caractères de contrôle du signal* qui sont des trames composées d'un seul octet de valeur 0XE5 envoyées dans le but de tester l'état de la communication(Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014). La Figure I.6 représente la structure générale des trames IEC-101.

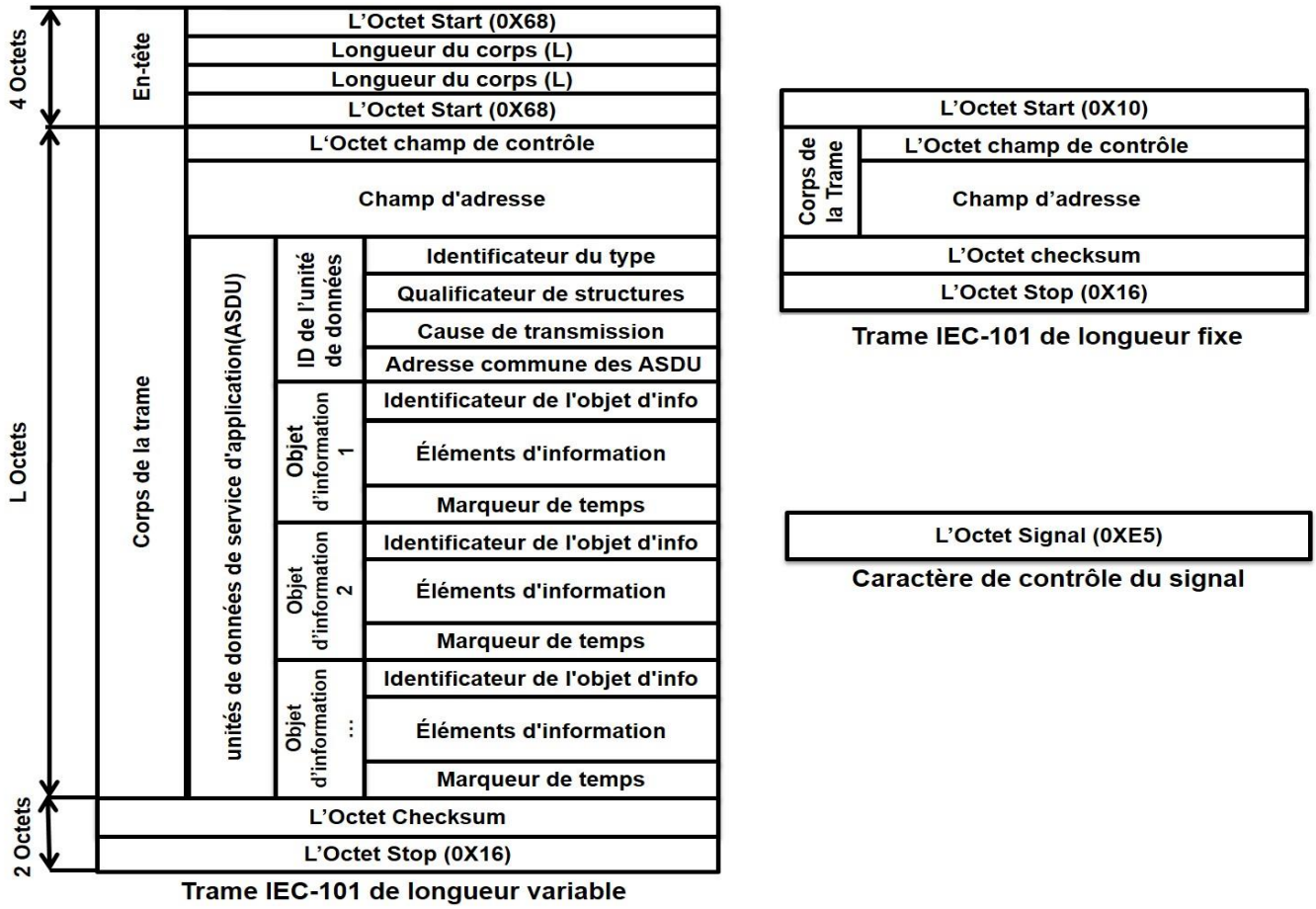
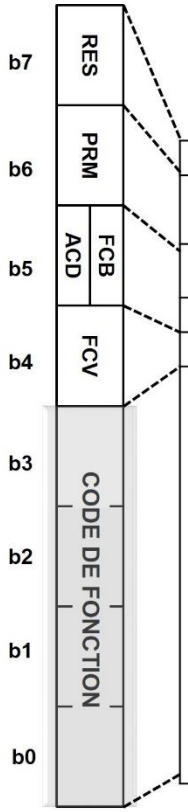


Figure I.6 Structure des trames IEC-101

L'en-tête des trames de longueurs variables est composé de quatre octets : Les deux octets (l'octet 1 et l'octet 4) sont des Byte-Start, ayant une valeur fixe égale à 0X68, qui indiquent aux récepteurs le format des trames envoyées. Le deuxième et le troisième octets ont la même valeur *L*. Ils indiquent la longueur du corps de la trame(Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014).

Le corps de la trame de longueur variable peut transporter jusqu'à 253 octets de données utilisateur. Sa longueur est indiquée par le deuxième octet de l'en-tête. Il est composé d'un octet de contrôle, d'un champ d'adressage et d'une unité de données de service d'application ASDU.

La valeur de l'octet de contrôle décrit la fonction de la trame. Elle dépend à la fois du sens (sens de contrôle/ sens d'affichage) et du mode de communication (communication symétrique/ communication asymétrique)(Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014). La Figure I.7 résume la signification de l'octet de contrôle en fonction du sens et du mode de communication.



		Maître vers esclaves		Esclave vers maître	
	VALEUR	Symétrique	Asymétrique	Symétrique	Asymétrique
RES	-	Réservé		Réservé	
PRM	0	Sens de communication inactif		Sens de communication actif	
	1	Sens de communication actif		Sens de communication inactif	
FCB	0-1	Changement de sa valeur entre 0 et 1 pour les trames séquentielles		//	
ACD	1	//		Présence de données classe 1	
FCV	1	Option du FCB validée		//	
CODE DE FONCTION	0	Initialiser la liaison		Confirmation - ACK	
	1	Reset le processus utilisateur		Confirmation - NASK	
	2	Tester la liaison	//	//	
	3	Données avec confirmation		//	
	4	Données sans confirmation		//	
	8	//		//	Réponse- Données utilisateur
	9	Demande de l'état de liaison		//	Réponse –Liaison NACK pas de données
	10	Demande données classe 1		//	
	11	Demande données classe 2		Réponse- Etat de Liaison	
	14	//		Liaison non opérationnelle	
	15	//		Liaison n'est pas utilisée	

Figure I.7 Signification de l'octet de contrôle dans une trame IEC 101

Le champ d'adressage (Address field bytes) peut être composé de zéro, un ou deux octets. Le protocole peut adresser 254 RTU lorsqu'il utilise un seul octet, 65534 RTU lorsqu'il utilise 2 octets, comme il peut être sans champ adresse si la communication est point à point. L'adresse 0 est réservée au maître alors que la dernière adresse « 0XFF dans le cas d'adressage par un octet et 0XFFFF dans le cas d'adressage par deux octets » est une adresse de diffusion.

Les en-queues des trames IEC-101 contiennent deux octets : L'octet de la somme de contrôle (Checksum), utilisé pour l'intégrité de la trame, suivi de l'octet d'arrêt (Stop-byte) de valeur 0X16.

D'autre part, chaque trame IEC-101 de longueur fixe est composée de l'octet de démarrage ayant la valeur 0X10, de l'octet de contrôle, du champ d'adressage, de l'octet de la somme de contrôle et de l'octet d'arrêt avec sa valeur de 0X16.

La couche Application du protocole IEC-101 présente une structure d'unités de données de service d'application ASDU très efficace. Chaque trame contient une seule et simple ASDU composée d'un identificateur d'unité de données (DUI - DATA UNIT IDENTIFICATION) et d'Objets d'information (voir Figure I.6).

Le Tableau I-1 représente une description des importantes valeurs des champs de l'ASDU de la trame IEC-101.

Tableau I-1 Description des importantes valeurs des champs de l'ASDU de la trame IEC-101

Champ ASDU		Longueur	Valeur	Description
Identificateur d'Unité de Données	**Identificateur du type**	01 Octet	3	Message Double
			4	Message double avec repère temporel
			5	Message de position prise
			13	Valeur de mesure, nombre à virgule flottante
			14	Valeur de mesure, nombre à virgule flottante avec repère temporel
			46	Commande double
			47	Commande de changement de prises
			50	Valeur de consigne de la commande de position, nombre à virgule flottante
			100	Commande de requête générale
			103	Commande de synchronisation temporelle
			112	Paramètre de valeurs de mesure
	Qualificateur de structure	01 Octet	Nombre	
	Cause de transmission	01 Octet	03	Spontanée
			06	Activation
			07	Confirmation de l'activation
			10	Fin de l'activation
			20	Demande par requête générale
	Adresse commune des ASDU	01 Octet	Réglable	
Objets d'information	**Adresse de l'objet d'information**	01/02 Octets	Réglable	
	Eléments d'information	N Octets	Réglable	

Chaque trame IEC-101 porte un seul type d'objets d'information. Chaque type d'objets d'information est définie par (1) son Identificateur du type, codé sur un octet et utilisé pour définir la structure, le type et le format des objets d'information ;(2) son Qualificateur de structures, codé sur un octet et utilisé pour indiquer le nombre d'objets ou d'éléments d'information dans la trame ; (3) sa Cause de transmission, codée sur un ou deux octets, qui définit l'objectif de l'envoi de la trame ; et finalement (4) son Adresse commune des ASDU, codée sur un ou deux octets, qui définit la station concernée par les objets d'information, généralement correspond à l'adresse de la RTU(Clarke et al. 2004; IEC_TC57 1995; Skoko et al. 2014).

Un objet d'information contient un ou plusieurs éléments d'information nécessaires à la téléconduite des équipements et des installations techniques. Sa structure, son type et son format sont définis par l'identificateur de type. Chaque objet d'information est composé, généralement, par (1) un Identificateur de l'objet d'information qui représente l'adresse de l'objet d'information ; (2) Un ensemble d'Éléments d'information qui contient les données d'information sur l'objet

d'information ; et d'une façon optionnelle (3) le champ Marqueur de temps (Time-tag) qui indique l'instant de la génération des données portées par les éléments d'information.

La simplicité et la robustesse de l'architecture EPA du protocole IEC60870-5-101 lui ont permis d'être adopté par plusieurs systèmes SCADA de nombreuses infrastructures critiques autres que l'industrie électrique. Elle sert comme plate-forme pour le développement d'autres protocoles de transport de données SCADA. Le protocole IEC 60870-5-104 est une adaptation du protocole IEC-101 aux réseaux routables. Il adopte les couches d'application et de liaison du protocole IEC-101, et intègre des services TCP/IP dans la couche physique du modèle EPA du protocole. Le protocole de transmission PUR2.4 est un autre protocole dérivé de l'IEC60870-5-101 qui permet la communication symétrique via des topologies Point à multipoints, anneaux ou bus par l'introduction de la méthode d'accès CSMA/CD et un meilleur contrôle de l'intégrité par l'usage du CRC-16 au lieu du Byte-Checksum (Smaiah et al. 2015b).

2. Le protocole DNP3

Le protocole DNP3 (Distributed Network Protocol version 3) est un protocole de transport de données SCADA ouvert très répandu pour la téléconduite de plusieurs infrastructures critiques en Amérique, Afrique du Sud, Asie, Australie et Nouvelle-Zélande. Basé initialement sur l'IEC 60870-5-1, Le WESTRONIC (actuellement connu par GE-HARRIS CANADA) l'a développé pour assurer une communication série fiable entre les MTU et les RTU, d'un côté, et entre les RTU et les IED d'un autre côté(Clarke et al. 2004; Gao et al. 2014; Stouffer et al. 2014).

Aujourd'hui, le DNP3 est suivi, maintenu et fait évoluer par le DNP3 User Group qui comporte les grands constructeurs et utilisateurs des solutions DNP3. Il est utilisé dans la gestion de nombreuses infrastructures critiques (la distribution de l'eau, le traitement des eaux usées, le transport et l'industrie des hydrocarbures...)

L'architecture du DNP3 lui permet d'exploiter à la fois des réseaux routables et des réseaux non-routables. Elle se base sur une architecture à performances améliorées (EPA) à trois couches composées de la couche physique, la couche liaison de données et la couche application. Une pseudo- couche supplémentaire, dite couche pseudo-transport, est introduite entre la couche liaison de données et la couche application afin de permettre la transmission des données d'importantes tailles. Les spécifications de ces couches ont été organisées sous forme de quatre documents de base

connus par (1) la Description de la couche liaison des données ; (2) les Fonctions du Transport ; (3) Description de la couche Application du protocole ; et (4) la librairie des objets de données. Un document supplémentaire est ajouté par la suite pour faciliter la conception et l'interopérabilité entre les systèmes des différents constructeurs. Il définit les classes des équipements DNP3 en précisant les types d'objets et les fonctions supportés par chaque classe (Clarke et al. 2004; Gao et al. 2014; Stouffer et al. 2014).

La Figure I.8 présente la structure de l'architecture EPA du protocole DNP3.

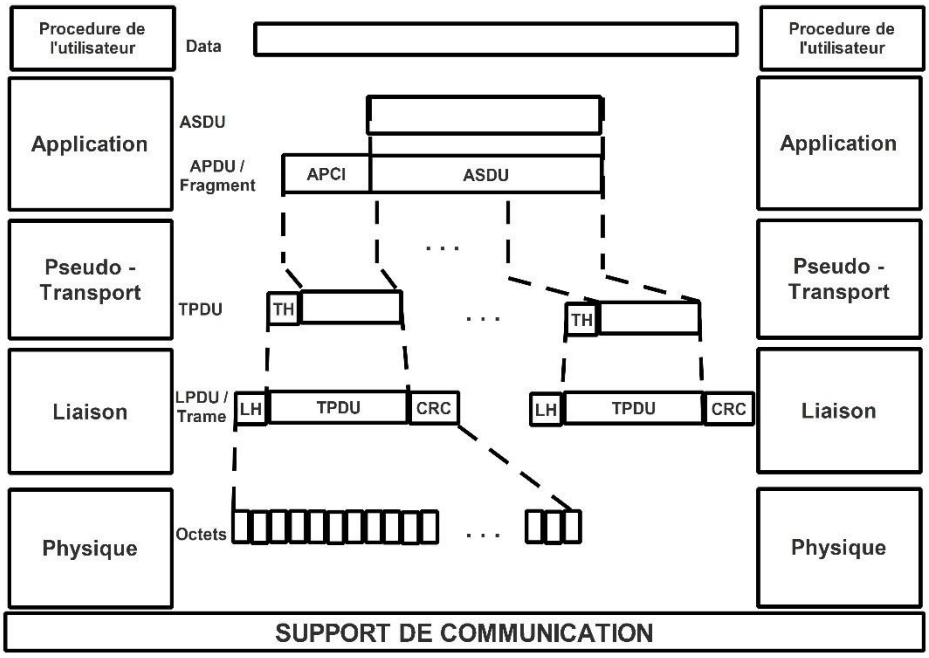


Figure I.8 Structure EPA du protocole DNP3

La couche physique du DNP3 autorise l'utilisation de toutes les technologies de communications séries routables, qui utilisent les fragments TCP/UDP et les paquets IP sur des liaisons Ethernet ou fibres optiques, et/ou non-routables qui fonctionnent avec des liaisons ITU-T V.24/ ITU-T V.28 ou des liaisons X.24 / X.27. Elle accepte, également, toutes les configurations de réseaux point à point, point à point multiple, multipoint en étoile, multipoint en ligne partagée et multipoint en anneau(Clarke et al. 2004; Gao et al. 2014; Stouffer et al. 2014).

La couche liaison de données permet d'utilisation des trois modes de communication à savoir (1) le mode symétrique ; (2) le mode asymétrique ; et (3) le mode multi-maîtres. Elle adopte le format de trame FT3 définit par l'IEC60870-5-1 qui peut adresser jusqu'à 65519 stations du système SCADA.

Elle transforme les unités de données de protocole de transport de la pseudo-couche Pseudo-transport en trames appelées Unités de Données de Protocole de Liaison (LPDU - Link Protocol Data Unit).

Chaque trame DNP3 est composée d'un en-tête et du corps de la trame. La Figure I.9 représente la structure de la trame DNP3 (Clarke et al. 2004; Gao et al. 2014; Stouffer et al. 2014).

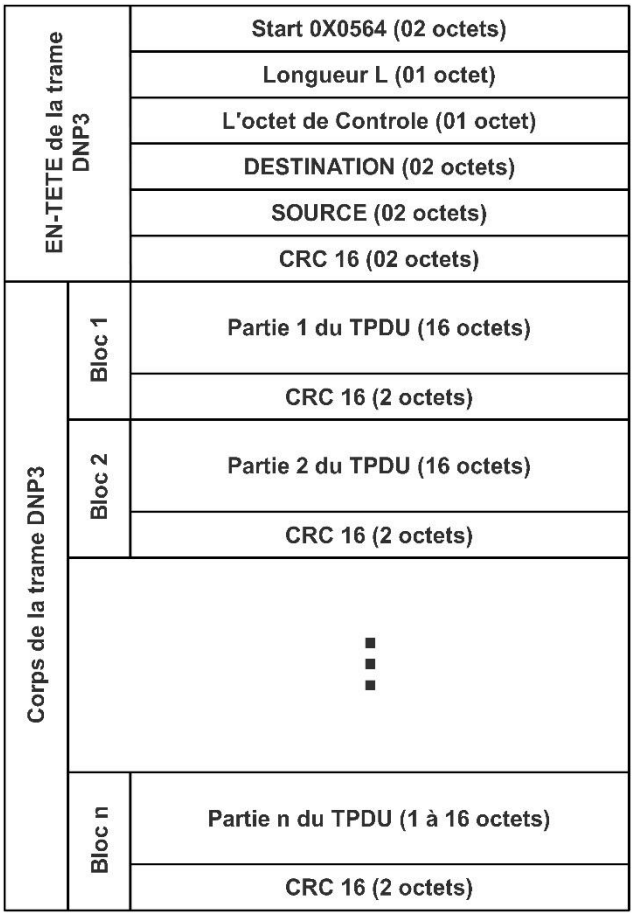


Figure I.9 Structure de la trame DNP3

L'en-tête de la LPDU contient six champs. (1) Le champ de démarrage est codé sur deux octets et de valeur fixe égale à 0X0564. Il indique le type de la trame ; (2) L'octet de *longueur de la trame L* indique la longueur du segment de données, dit Unités de Données du Protocole de transport (TPDU - Transport Protocol Data Unit) transporté par la trame ; (3) L'*octet du Contrôle* similaire à l'octet de contrôle des trames IEC 60870-5-101. Il décrit la fonction de la trame. Elle dépend à la fois du sens et du mode de communication. La signification de sa valeur en fonction du sens et du mode de la communication a été déjà décrite dans la Figure I.7. (4) Le *champ Destination* est codé sur deux

octets. Il définit l'adresse du récepteur destinataire de la trame ; (5) Le *champ Source* est codé sur deux octets. Il définit l'adresse de l'émetteur de la trame ; (6) Le champ *Contrôleur de redondance cyclique* à 16 bits de l'en-tête (CRC - Cyclic Redundancy Check) est codé sur deux octets. Il est calculé sur les huit octets précédents afin de détecter toute modification sur l'en-tête en se basant sur le polynôme :

$$P(x)= X^{16}+ X^{13}+ X^{12}+ X^{10}+ X^{8}+ X^{5}+ X^{2}+ 1. \quad (1.1)$$

La couche liaison découpe la TPDU en n parties, les (n-1) premières parties ont une taille de 16 octets, la dernière partie contient le reste de la trame TPDU. Le corps de la trame est constitué de n blocs, chaque bloc est composé d'une partie de TPDU suivie de deux octets contenant son CRC-16.

La pseudo-couche Pseudo-transport comporte des fonctions de fragmentation et d'assemblage des données. Elle forme les unités de données de protocole de transport (TPDU - Transport Protocol Data Unit) en découpant l'unité de données du protocole d'Application (APDU- Application Protocol Data Unit) en groupes de 249 octets chacune. Elle ajoute à chaque groupe un en-tête transport, d'un octet, pour indiquer sa position dans l'APDU.

La couche Applicative du DNP3 offre une large gamme de fonctions possibles qui peuvent se distinguer en deux catégories : les fonctions applicatives et les fonctions du système. Les fonctions applicatives contiennent l'ensemble des fonctions d'accès génériques aux données d'équipement esclave, transmission de commandes, de fichiers ou d'événements horodatés, gestion des compteurs ou des programmes... Les fonctions du système comportent les fonctions de synchronisation du temps, redémarrage du système et configuration du mode de communication (Clarke et al. 2004; Gao et al. 2014; Stouffer et al. 2014)...

D'autre part, le DNP3 expose une riche variété d'objets qui offre une meilleure représentation des données échangées classifiées en trois types. Le premier type est les objets binaires tels que les Binary Inputs, les Binary outputs, les Binary Input change events, les control Relay Output Blocks... Le deuxième représente les objets analogiques dont on trouve les Analog Inputs, les Analog Outputs, et les Analog Input change Event. Le dernier type est les compteurs.

La couche Application traduit les données d'utilisateurs en plusieurs unités de données du protocole d'application (APDU - Application Protocol Data Units) d'une taille maximale de 2048 octets. Chaque APDU est construite d'un en-tête dit : Information de Contrôle du Protocole d'Application

(APCI - Application Protocol Control Information) et d'une unité de données de service d'application (ASDU - Application Service Data Units). Dans le cas d'un envoi d'une commande ou d'une demande sans transmission de données, l'APDU ne contient que l'APCI (Clarke et al. 2004; Gao et al. 2014; Stouffer et al. 2014).

L'APCI est construite, dans le cas d'une réponse à une interrogation, de quatre octets : (1) L'octet Contrôleur d'application qui exprime le sens de la communication, la position du fragment dans les données de l'utilisateur et le besoin d'accuser la bonne réception de l'APDU ; (2) L'octet Code de la fonction d'application qui indique la fonction à exécuter ; et (3) L'Indicateur interne composé de deux octets et dont chaque bit de ce champ représente un flag indiquant (1)l'état de la réception des intégrations, (2) la classe des données transmises, (3) l'utilisation d'une synchronisation temporelle, (4) les anomalies et des erreurs de réception des messages ou de fonctionnement des équipements.

Dans le cas d'une interrogation, L'APCI n'est composée que de l'octet du contrôleur d'application et de l'octet du code de la fonction d'application.

L'ASDU du DNP3 est composée d'un ou plusieurs objets d'information. Le nombre d'objets d'information détermine la taille de l'APDU, qui ne doit pas dépasser les 2048 octets de taille. Chaque Objet d'information est composé d'un en-tête d'objet et d'un champ d'objet contenant l'ensemble des données définies par l'en-tête d'objet. L'en-tête d'objet identifie le type de l'objet de données et les instants spécifiques des données référencées par le message. Il est composé de trois champs à savoir : (1) Le champ Objet, codé sur deux octets, qui détermine le type des données contenues dans le champ d'objet de données (binaire, analogique, compteur...) ; (2) L'octet Qualificateur d'objet utilisé pour spécifier les points de données spécifiques ; et (3) Le Rang de l'objet, ayant une taille variable de zéro octet à huit octets, qui complète l'information fournie par le Qualificateur d'objet tel que les instants spécifiques des données(Clarke et al. 2004; Gao et al. 2014; Stouffer et al. 2014).

Suite à la richesse de la couche applicative en fonctions et en types d'objet, le document référentiel « DNP3 SubsetDefinitions » a distingué les équipements DNP3 en trois classes en fonction des éléments utilisés. (1) Le DNP-L1 est la classe adressée aux équipements les plus basiques utilisant un ensemble d'éléments très réduit tels que certains appareils de mesures. (2) Le DNP-L2 est adressée à la gestion des équipements de niveau intermédiaire tels les PLC, les IED et les petites RTU. Enfin, (3) le DNP-L3 est le niveau le plus haut qui permet la gestion de tous les équipements

de télécontrôle tels les calculateurs, les concentrateurs et les RTU(Clarke et al. 2004; Gao et al. 2014; Stouffer et al. 2014).

3. Le Modbus

Le MODBUS, acronyme de MODicon communication BUS, est probablement le protocole de communication asymétrique (maître-esclave) le plus ancien et le plus utilisé dans les communications des systèmes de contrôle distribués. En raison de sa popularité et de la facilité avec laquelle le Modbus peut être mis en œuvre, le protocole a été adopté par de nombreux équipements et adapté à de nombreux systèmes industriels, notamment les systèmes SCADA. Aujourd'hui, Le Modbus trouve sa place dans la liste des protocoles de transport SCADA même s'il est conçu initialement comme un protocole de terrain. Non seulement les dispositifs intelligents puissants tels que les RTU et les PLC peuvent communiquer avec le Modbus, mais de nombreux capteurs intelligents disposent d'une interface Modbus qui leur permet d'envoyer leurs données aux systèmes hôtes (Erez and Wool 2015; Goldenberg and Wool 2013; Huitsing et al. 2008; ModbusIDA 2004).

La popularité du Modbus a augmenté en raison de sa structure de messagerie simple et efficace. Du point de vue architecturale, le Modbus est classé, selon le modèle TCP/IP, comme un protocole de la couche applicative. Il présente une structure indépendante du type de l'interface physique et des connexions qui lui a permis de présenter plusieurs variantes adaptées à chaque interface. Son unité de données de service d'application (ASDU) est composée de deux champs : Le code de fonction et la gamme de données(Knapp and Langill 2014; ModbusIDA 2004).

Le code de fonction détermine la façon dont le maître peut accéder aux données et les modifier. Contrairement aux gammes de données, qui sont conceptuelles, les codes de fonction ont un comportement bien défini. Ils sont définis dans la plage décimale 1-255 (le code 0 n'est pas valide) où on distingue trois catégories des codes de fonction : (1) Les codes de fonction publique, définis sur les plages 1-64, 73-100 et 111-127, et qui sont uniques, bien déterminés, documentés et validés par la communauté MODBUS.org; (2) Les codes de fonction définis par l'utilisateur, définis sur les plages 65-72 et 101-110 et permettant aux utilisateurs de sélectionner et de déterminer des codes de fonction qui ne sont pas pris en charge par la spécification; et finalement (3) les codes de fonction réservés, définis de la plage 128-255, qui sont réservés aux produits de certaines entreprises sans être disponibles pour l'utilisation publique(ModbusIDA 2004).Le Tableau I-2 représente une description des codes de fonction les plus populaires.

Tableau I-2 Description des codes de fonction les plus populaires.

Code fonction	Description des fonctions Modbus	Code fonction	Description des fonctions Modbus
0X01	Lecture de n bits de sortie consécutifs.	0X0B	Lecture du compteur d'événements
0X02	Lecture de n bits d'entrée consécutifs.	0X0C	Lecture des événements de connexion
0X03	Lecture de n mots de sortie consécutifs.	0X0D	Téléchargement / télédéchargement
0X04	Lecture de n mots d'entrée consécutifs.	0X0E	Demande de CR de fonctionnement.
0X05	Écriture de 1 bit de sortie.	0X0F	Écriture de n bits de sortie.
0X06	Écriture de 1 mot d'entrée.	0X10	Écriture de n mots de sortie.
0X07	Lecture du statut d'exception.	0X11	Lecture d'identification.
0X08	Accès aux compteurs de diagnostic.	0X12	Téléchargement / télédéchargement.
0X09	Téléchargement / télédéchargement	0X13	Reset de l'esclave après erreur non recouverte.
0X0A	Demande de CR de fonctionnement.		

Les données transmises par le Modbus sont organisées dans des registres à adresses prédéfinies. Quatre formats de données sont supportés par la norme, à savoir : (1) Les bobines qui sont des données définies en mode lecture / écriture, codées sur un bit et adressées dans la plage 00001-09999 ; (2) Les entrées discrètes qui sont des données définies en lecture uniquement, codées sur un bit. Elles sont adressées dans la plage 10001-19999 ; (3) Les registres d'entrées qui sont des données en lecture uniquement, codées sur deux octets et adressées dans la plage 30001-39999 ; finalement (4) Les registres de maintien qui sont des données, en lecture / écriture, codées sur deux octets et adressées dans la plage 40001-49999 (ModbusIDA 2004).

La couche application forme l'unité de données du protocole d'application APDU en ajoutant à l'ASDU un en-tête APDU, qui contient des informations sur l'adresse de l'esclave, et une en-queue APDU qui contient un champ détecteur d'erreur (ModbusIDA 2004).

L'implémentation du protocole Modbus sur plusieurs interfaces de niveaux inférieurs a permis la définition de plusieurs variantes Modbus classées en deux familles : La famille des protocoles Modbus non routables et la famille des protocoles Modbus routables(ModbusIDA 2004). La Figure I.10 représente la structure protocolaire, selon le modèle TCP/IP, des deux familles de variantes Modbus.

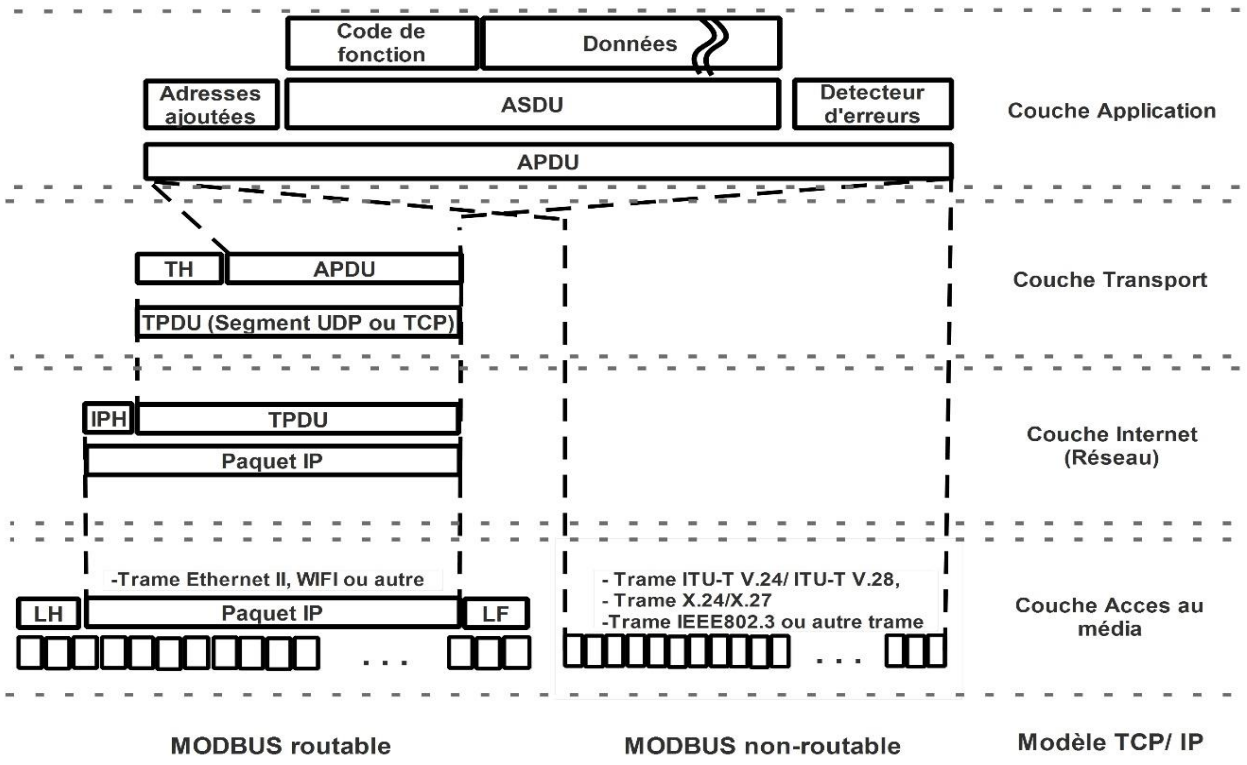


Figure I.10 Structure protocolaire des variantes Modbus

Les variantes MODBUS non-routables ne sont composées que des deux couches basiques : La couche Accès au media et La couche application définie précédemment. La couche Accès aux media définit l'interface physique. Elles utilisent généralement des protocoles de transmission série asynchrone tels EIA/TIA-232-E, EIA-422, EIA / TIA-485-A, IEEE802.3, fibre... Nous distinguons deux variantes principales de cette famille : Le Modbus RTU et le Modbus ASCII.

Le Modbus RTU impose un codage en mode binaire des champs de la trame. L'en-tête est composé de l'adresse de l'esclave codée sur un octet alors que l'en queue de la trame est composé de deux octets contenant la valeur du CRC-16 de la trame. La fin d'émission d'une trame est traduite par un silence de durée supérieure à la durée d'émission de trois octets(ModbusIDA 2004).

Le Modbus ASCII code les champs de la trame en mode ASCII où chaque octet représente un caractère de la trame). L'en-tête de la trame est composé d'un octet de démarrage (Start Byte) de valeur 0X3A et de l'adresse de l'esclave codée sur deux caractères. L'en-queue est composée du LRC (Longitudinal Redundancy Check) de la trame, codé sur deux caractères, et des deux octets de fin de la trame portant la valeur 0X0D0A (ModbusIDA 2004).

Les variantes Modbus routables exploitent toutes les couches du modèle TCP/IP. Nous citons trois importantes variantes ouvertes routables du Modbus : Le Modbus sous TCP, le Modbus TCP et le Modbus UDP. Le Modbus sous TCP, appelé également Modbus RTU/IP, embarque L'APDU du Modbus RTU sur une trame TCP/IP en utilisant le port TCP 502. Le Modbus TCP ressemble au Modbus RTU/IP avec une petite différence au niveau de l'APDU embarquée, il utilise un adressage IP au lieu de l'adressage RTU et son APDU ne contient pas une en-queue car la détection des erreurs est assurée par les couches inférieures. Finalement, Le Modbus UDP embarque L'APDU du Modbus RTU sur une trame UDP/IP.

VI. Conclusion

Nous avons présenté, le long de ce chapitre, des notions de base sur les systèmes SCADA, leurs évolutions ainsi que les architectures matérielles et protocolaires des systèmes SCADA. Nous avons défini également les trois principaux protocoles de transport SCADA utilisés dans le domaine de la téléconduite des réseaux (IEC 60870-5-101, DNP3 et le Modbus). La comparaison entre ces trois protocoles montre que :

Malgré la réputation du protocole Modbus dans les communications industrielles, son usage reste limité dans les applications SCADA de téléconduite. En effet, le Modbus est pauvre en fonctionnalité et en types d'objets vue sa conception originale orientée réseaux de terrain.

Nous pouvons remarquer que l'IEC 60870-5-101 offre une plus grande flexibilité dans le routage des messages. En effet, il utilise deux types d'adresses : l'adresse de liaison définissant la station et l'adresse d'application définissant l'objet alors que le DNP3 n'utilise que des adresses liaisons. D'autre part, L'IEC-101 utilise des longueurs variables d'adresses, ce qui permet d'économiser la bande passante de communication. Comme il peut communiquer en mode symétrique ou asymétrique contrairement au DNP3 qui ne communique qu'en mode symétrique.

Le DNP3 est conçu pour supporter une configuration symétrique. Par contre, la communication symétrique dans le protocole IEC 60870-5-101 standard est limitée à la topologie point-à-point. Le PUR2.4 est un protocole dérivé de l'IEC-101 étendue à la topologie multipoint grâce au mécanisme CSMA/CD ajouté au protocole pour gérer les collisions. Ceci signifie que, dans la situation d'une communication point multipoint avec un très grand nombre de stations et une bande passante limitée, le DNP3 et le PUR2.4 sont les meilleurs choix par rapport à l'IEC 60870-5-101.

Du point de vue format de trame, les protocoles IEC 60870-5-101 utilisent des trames FT1.2, de longueurs fixe et variable, alors que le DNP3 utilise une trame FT3 de longueur variable seulement. L'option de longueur fixe réduit considérablement les frais généraux de communication.

Le IEC 60870-5-101 utilise un octet de somme de contrôle pour une longueur de trame de 255 octets ; le PUR-2.4 utilise deux octets CRC et un octet de somme de contrôle pour une longueur de trame de 255 octets au maximum et le DNP3 utilise deux octets CRC pour une longueur de trame de 255 octets. En conséquence, la détection d'erreur est plus forte dans les protocoles PUR2.4 et DNP3. Pour les fonctions d'application, l'IEC 60870-5-101 n'autorise qu'un seul point de contrôle par message (une fonction par trame), mais le DNP3 permet le contrôle à travers plusieurs points dans un seul message (plusieurs fonctions par trame).

Les protocoles SCADA présentés sont des protocoles temps réel efficaces non-gourmands en ressources. Nous les avons implémenté sur des cartes de prototypage rapide à ressources limitées de type Arduino-Uno et Raspberry-Pi II (Smaiah et al. 2015a; Smaiah et al. 2015b). Plusieurs autres implémentations et bibliothèques sont fournies, gratuitement, par les constructeurs des solutions amateurs et professionnels sur internet.

D'autre part, l'utilisation de ces protocoles ouverts sur des réseaux publics de communication présente une grande menace sécuritaire sur les systèmes contrôlés. En effet, les protocoles SCADA modernes ne présentent aucune protection face aux attaques cybernétiques. L'utilisation des algorithmes d'authentification et de cryptographie standards avec des équipements de ressources limitées peut affecter le fonctionnement temps réel du système SCADA. Nous présentons dans le Chapitre II un aperçu général sur l'état sécuritaire des systèmes SCADA modernes en se basant sur l'analyse des cyber-attaques les plus destructives dans l'histoire des réseaux de contrôle industriels.

CHAPITRE II : APERCU SUR LA SECURITE DES SYSTEMES SCADA

I. Introduction

L'introduction des systèmes informatiques temps réel pour le contrôle des processus industriels a participé vivement à l'évolution de différents secteurs stratégiques. En effet : Les systèmes SCADA actuels fournissent plusieurs services indispensables : Ils assurent une surveillance et un contrôle, en temps réel, de la quasi-totalité des infrastructures critiques modernes. Ils peuvent offrir des analyses instantanées sur les événements produits ; archiver des données importantes ; générer des rapports détaillés ; aider à la simulation des événements en utilisant des paramètres réels et collaborer avec d'autres systèmes dans l'élaboration et la mise à niveau des stratégies des entreprises...

Toutes ces fonctionnalités et d'autres rendent la gestion des infrastructures critiques fortement dépendante des systèmes SCADA. Tout incident ou dysfonctionnement de ces systèmes peut entraîner d'importantes conséquences humaines, politiques, stratégiques, gouvernementales, sociales, économiques ou environnementales. De ce fait, les systèmes SCADA font partie des champs de bataille invisibles, les plus attirants, dans la courante guerre cybernétique.

Les cyber-attaquants visent à contrôler ces réseaux critiques, à les perturber ou, dans un niveau plus bas, à accéder à leurs données stratégiques. Ils exploitent leur grande connaissance des vulnérabilités des protocoles ouverts utilisés, des réseaux de télécommunication publics, des connectivités, des stratégies de la sécurité IT, des architectures matériels et logiciels des réseaux pour organiser des cyber-attaques efficaces (Nicholson et al. 2012; Robinson et al. 2015).

La sécurité des systèmes SCADA se base essentiellement sur la disponibilité, l'intégrité, la confidentialité, l'authentification, la traçabilité et l'imputation des données communiquées. La disponibilité signifie que le système peut garantir un fonctionnement continu normal sans interruption ou arrêt. L'intégrité garantie la réception des données émises sans modifications (ni

variation ni manque).La confidentialité assure le secret des messages communiqués, aucun étranger ne peut interpréter les informations portées par ces messages. L'authentification indique la distinction et l'identification de la source de chaque message alors que la traçabilité traduit la connaissance et la conservation de toutes les informations sur l'activité et le trafic du système. Finalement, l'imputation assure que chaque intervenant ne peut effectuer que les actions dont il possède leurs autorisations.

II. **Comparaison entre la sécurité des systèmes SCADA et des technologies d'information (IT)**

Afin de sécuriser les systèmes SCADA, les concepteurs potentiels des réseaux de contrôles introduisent généralement des solutions sécuritaires conçues pour les systèmes informatiques IT. Les pare-feu standards, les outils d'authentification usuels, les algorithmes de cryptage de même que les autres solutions IT ne sont pas assez performants pour sécuriser des réseaux industriels contre les cyber-attaques. En effet, même si les réseaux industriels ont bien profité dans leurs développements des technologies des réseaux d'information, la sécurisation des systèmes SCADA requière la satisfaction de certaines exigences spécifiques différentes de celles exigées par les systèmes IT. En effet, les exigences concernant la qualité de service de chaque type des deux réseaux, les objectifs et les architectures sécuritaires ainsi que les technologies matérielles et logicielles sont tous différents (Lu et al. 2010; Wei et al. 2010; Zhu et al. 2011).

Les systèmes SCADA sont des réseaux industriels qui gèrent des processus vitaux des infrastructures critiques. Chacune de ses perturbations fonctionnelles peut potentiellement entraîner des conséquences significatives sur le monde physique. Contrairement à la majorité des systèmes d'information qui tolèrent des retards et des défaillances occasionnels, les systèmes de contrôle industriels exigent un très haut niveau de la qualité de service afin d'assurer un fonctionnement temps réel strict avec une disponibilité continue. En effet, chaque retard dans l'acquisition, le traitement, la transmission, l'analyse ou l'exploitation des données échangées et chaque interruption, redémarrage ou réinitialisation des processus ne peut être admis suite aux graves conséquences qui peuvent se répercuter sur l'état des infrastructures(Zhu et al. 2011).

Les objectifs sécuritaires de chaque type de systèmes est différent de l'autre. En effet, les systèmes d'information (IT) visent à sécuriser les données en assurant leurs intégrité, confidentialité,

authentification et disponibilité. Les systèmes de contrôle industriels visent principalement à sécuriser les services qui garantissent la protection des vies humaines, des installations et du fonctionnement normal des systèmes(Wei et al. 2010).

L'architecture sécuritaire des systèmes IT est différente de celle des systèmes de contrôle industriel. En effet, pour les systèmes IT, la protection des données est l'objectif sécuritaire le plus important. Par conséquent, les centres de données (Data centers), contenant les importants serveurs de données, exigent l'application d'un niveau de sécurité plus haut que celui appliqué sur les autres nœuds des réseaux. Concernant les systèmes de contrôle industriels, chaque partie des systèmes doit être hautement sécurisée : Les importants services sont gérés par les segments principaux des réseaux SCADA (Network data centers), qui communiquent en utilisant les segments de réseaux de coopération, et sont exécutés par les segments des équipements de terrain (Edge nodes)(Lu et al. 2010; Wei et al. 2010).

D'un autre point de vue, les centres de données des systèmes d'information IT sont installés dans des locaux dédiés avec des accès physiques limités. Contrairement aux systèmes de contrôle industriels, où les nœuds extrêmes (segment des équipements de terrain) peuvent être publiquement accessibles. En effet, les RTU installées dans des endroits accessibles non gardés, comme les lignes électriques, sur les autoroutes ou sur les chemins de fer, peuvent être favorisées pour lancer des cyber-attaques de différents niveaux(Shakarian et al. 2013).

Les systèmes d'exploitation, les interfaces de communication MMI, les systèmes de gestion de base de données ainsi que les protocoles de communication sont de simples exemples des technologies logiciels faisant la différence entre les systèmes IT et les systèmes ICS. Habituellement, les systèmes IT explorent des serveurs puissants utilisant des logiciels plus adaptés à la sauvegarde, la gestion, l'exploitation et la présentation de leurs données volumineuses. D'autre part, les réseaux de contrôle industriels utilisent des systèmes d'exploitation temps réel (RTOS- Real Time Operating Systems) en présence de ressources limitées. Ils doivent exploiter des technologies logicielles efficaces et optimisées qui s'adaptent mieux à leurs architectures matérielles et qui assurent un bon fonctionnement temps réel avec une faible consommation énergétique (Lu et al. 2010; Wei et al. 2010).

Finalement, les composants des systèmes IT ont une durée de vie moyenne qui varie entre trois et cinq ans. Plusieurs vulnérabilités peuvent être traitées durant cette période par de simples évolutions

technologiques ou développement de nouvelles mises à jour. Par conséquent, le renouvellement et l'actualisation de l'équipement augmente la résistance du système aux cyber-attaques. Ceci ne semble pas évident avec les systèmes de contrôle industriels qui utilisent à la fois des équipements coûteux et d'une longue durée de vie qui peut dépasser les vingt ans (Shakarian et al. 2013).

III. Principaux auteurs des cyber-attaques visant les systèmes SCADA

L'analyse de l'effet des systèmes SCADA sur le fonctionnement des infrastructures critiques nous a montré que ces dernières peuvent être fortement menacées par la production de réels accidents suite à de mauvaises manipulations, des incidents ou des dysfonctionnements d'une partie ou de la totalité des systèmes SCADA. Le niveau de ces accidents peut varier du simple, sans influences significatives, à catastrophique irréversible(Miller and Rowe 2012).

Du point de vue historique, avant l'an 2000, plus de 70% des incidents SCADA signalés étaient de sources internes. Elles étaient causées principalement par des erreurs de manipulation, des anomalies de conception des systèmes ou dues à des actes de sabotage ou de vandalismes intentionnels. Depuis 2001, le nombre d'incidents de grandes ampleurs n'a cessé d'augmenter (09 incidents en 2009, 39 incidents en 2010, 198 incidents en 2011 rien qu'aux Etats Unis d'Amérique (Knowles et al. 2015)) où plus de 70% d'entre eux sont dus à des cyber-attaques de sources externes (Alcaraz and Zeadally 2015; Igure et al. 2006). La principale explication de cette inversion des statistiques est l'évolution des systèmes SCADA vers la troisième génération utilisant des protocoles ouverts, exploitant des réseaux publics et autorisant les échanges d'information avec les autres systèmes coopératifs. Au jour d'aujourd'hui, nous comptons sept auteurs principaux des cyber-attaques visant les systèmes SCADA(Gao et al. 2014; Miller and Rowe 2012; Nicholson et al. 2012) :

1. Les employés mécontents :

Statistiquement parlant, la majorité des cyber-attaques ciblant les systèmes SCADA étaient d'origines internes. Les employés mécontents sont les auteurs historiques des cyber-attaques. Ils ont, en plus de la motivation et la facilité, la couverture pour effectuer de telles attaques sans être identifiés. En effet, ils ont le pouvoir d'accéder physiquement aux systèmes en contournant efficacement toutes les procédures de sécurité mises en œuvre basées essentiellement sur le contrôle d'accès, les pare-feu et l'analyse du trafic.

En 1992, un employé licencié du réseau d'alerte de la deuxième compagnie pétrolière américaine Chevron (Chevron Emergency network) a désactivé le système d'alerte de l'entreprise. Le vandalisme n'a été découvert qu'après une dizaine d'heures de la libération d'une substance nocive de la raffinerie de Chevron, située à Richmond (en Californie). Par conséquent, des milliers de personnes dans vingt-deux États et six régions non précisées du Canada ont été mises en péril alors que le système était en panne(Denning 2000).

En 1999, des pirates, en collaboration avec un employé mécontent, ont utilisé un cheval de Troie pour prendre le contrôle de la MTU, qui contrôle le flux et le débit de gaz dans les pipelines du Gazprom, une compagnie de gaz en Russie(Denning 2000).

Huit ans plus tard, en Californie, un ancien superviseur électrique mécontent de Tehama Colusa Canal Authority (TCAA) a installé le jour de son licenciement un malware sur le système SCADA. Alors que l'employé était condamné à 10 ans de prison et une amende de 250.000 dollars, aucun rapport technique ou analyse n'a été rendu public concernant le logiciel ou ses dommages causées (Nicholson et al. 2012).

2. Les mauvaises manipulations

Les mauvaises manipulations, les erreurs de conception et les actes de maintenance peuvent produire de véritables menaces. En juin 1999, une défaillance d'un pipeline de 16 pouces a causé l'écoulement de 900 mètre cube d'essence dans le ruisseau qui a traversé Whatcom Falls Park à Bellingham, Washington. Environ une heure et demie après la rupture, l'essence s'est enflammée où il a brûlé, à environ 1/2 mile le long du ruisseau, en causant 3 décès et 8 blessures. Le rapport publié en octobre 2002 par le National Transportation Safety Board (NTSB) mentionnait que l'une des cinq principales causes de l'accident était la modification des bases de données sur le système SCADA suite à des travaux de maintenance. Bien qu'on ne connaisse pas si l'incident est dû à une cyber-attaque ou pas, la perte de vies humaines dans cet incident illustre les dangers de tout type de défaillance dans un système d'infrastructure critique(Tsang 2010).

3. Les pirates étatiques

Les pirates étatiques sont des hackers engagés par les gouvernements comme membres de leurs armées d'élites formées pour la cyber-guerre. Au jour d'aujourd'hui, plus de 120 pays, y compris l'Algérie, ont initié des stratégies pour se protéger face aux attaques cybernétiques.

Les hackers étatiques sont considérés comme l'une des menaces les plus dangereuses pour les systèmes SCADA. Leur financement par les gouvernements signifie que d'importants moyens sont mis à la disposition des plus brillants pirates embauchés afin d'organiser de nouvelles cyber-attaques destructives difficiles à détecter.

En 1982, des intrus ont planté un cheval de Troie dans le système SCADA qui contrôle le pipeline de Sibérie. Cela a provoqué une explosion équivalente à 3 kilotonnes de TNT. Cet événement est considéré comme le premier incident cyber-sécuritaire signalé impliquant une infrastructure critique(Daniela 2011). Suivant une version américaine de Thomas C. Reed, le 11ème secrétaire des forces aériennes américaines, l'explosion est dû à l'implémentation d'un cheval de Troie conçu par les américains sur le systèmes SCADA contrôlant un gazoduc que l'Union soviétique a obtenu d'une société Canadienne (Markoff October 26, 2009).

En 2009, des espions chinois et russes ont pénétré dans le réseau électrique américain, laissant derrière eux des logiciels potentiellement perturbateurs (Gorman 2009; Heginbotham et al. 2015). Un haut responsable américain a déclaré que les espions chinois et russes avaient tenté de cartographier les infrastructures critiques des États-Unis en utilisant des outils de cartographie réseau(Gorman 2009; Heginbotham et al. 2015).

En juin 2010, le ver informatique STUXNET a été identifié par les leaders de la sécurité informatique : Kaspersky (Russie), Symantech (USA) et VirusBlokAda (Biélorussie). Le ver, développé par le NSA (USA) en collaboration avec l'unité 8200 (Israël), s'agissait de la première arme cybernétique visant les systèmes SCADA. Il cible les systèmes SCADA WinCC/PCS 7 de Siemens comme ceux utilisés par le contrôle des centrifugeuses iraniennes d'enrichissement d'uranium. Composé d'un total de quatre vulnérabilités zéro-day(Falliere et al. 2011), la complexité de STUXNET a mis en évidence les véritables menaces de la Cyber-guerre(Kushner 2013). La cartographie de la propagation de ce ver montre qu'il a été fortement déployé en Iran et que l'un des scénario le plus probable était son introduction au système SCADA par l'usage d'un disque USB infecté(Erdbrink and Nakashima 28/09/2010). Cette attaque a indubitablement conduit les pays à considérer sérieusement la posture de la sécurité de leurs infrastructures critiques (Farwell and Rohozinski 2011; Kriaa et al. 2012; Langner 2011; Nicholson et al. 2012).

En 2011, la garde républicaine Iranienne a pu prendre le télécontrôle du RQ-170 sentinelle, le plus moderne drone furtif américain, et le faire attirer sur son sol en perturbant sa communication avec la

station de contrôle et en piratant son système de géolocalisation GPS(Hartmann and Giles 2016; Hartmann and Steup 2013). Avec l'aide des chinois et des russes, l'Iran a pu découvrir tous les secrets du drone. En 2014, L'Iran a offert à la Russie un exemplaire d'une version iranienne du drone pour montrer sa production massive de l'appareil.

En 2011, une nouvelle forme de logiciel malveillant, appelé DUQU, s'inspirant de plusieurs techniques de Stuxnet a été découverte. DUQU n'était pas un malware auto-repliable et ne contenait pas de charge utile. Il semble être conçu pour effectuer une reconnaissance sur un système de contrôle industriel non identifié(Chien et al. 2012; Faisal and Ibrahim 2012).

En février 2011, McAfee a rapporté que cinq entreprises énergétiques et pétrolières mondiales étaient ciblées par une combinaison d'attaques comprenant l'ingénierie sociale, le spear-phishing et les trojans(Heginbotham et al. 2015; Leyden 2011). Les attaques, connues sous « Night Dragon», ont été confirmées comme étant en cours depuis plus de deux ans et seraient d'origine chinoise. Cependant, les attaquants n'ont utilisé que des outils et des machines chinois afin de masquer leur identité. Les attaquants ont pu avoir des données stratégiques telles que des plans opérationnels(Cyberattacks 2011; Keizer 2011; Liu et al. 2012).

En 2012, Les analyseurs de réseaux ont découvert un autre malware qui active en Iran, au Liban, en Syrie, au Soudan, en Cisjordanie et dans d'autres régions du Moyen-Orient et d'Afrique du Nord. Ce malware surnommé "Flame" semble être sponsorisé par les créateurs de Stuxnet. Les premières analyses indiquent qu'il est conçu principalement pour espionner les utilisateurs d'ordinateurs infectés et pour voler des données, y compris des documents, des conversations enregistrées et des frappes au clavier. Il ouvre également une porte aux systèmes SCADA infectés pour permettre aux attaquants de modifier les boîtes à outils et d'ajouter de nouvelles fonctionnalités. Flame a été découvert après que l'Union Internationale des Télécommunications des Nations Unies ait demandé aux chercheurs d'examiner les rapports du mois d'avril selon lesquels des ordinateurs appartenant à la compagnie iranienne du Pétrole (Iranian National Oil Co) avaient été infectés par des logiciels malveillants.

3. Les hacktivistes et les cyber-terroristes

Les activistes hackers ou «hacktivistes» sont des militants informaticiens qui militent pour les différentes causes géopolitiques, sociales, religieuses, ou environnementales nobles. Ils utilisent le cyberespace pour manifester et réclamer leurs révocations. Alors que dans le passé proche, des

manifestations physiques exprimaient leurs positions. Aujourd'hui, des protestations similaires pourraient avoir lieu en attaquant des systèmes informatiques SCADA à partir de n'importe quel point du monde connecté(Nicholson et al. 2012).

La définition du cyber-terrorisme, selon Dorothy E. Denning de Georgetown University, est : «l'ensemble des attaques informatiques ou des menaces d'attaques par des acteurs non étatiques contre les systèmes d'information lorsqu'ils sont menés pour intimider ou contraindre des gouvernements ou des sociétés à poursuivre des objectifs politiques ou sociaux. C'est la convergence du terrorisme avec le cyberespace, où le cyberespace devient le moyen de mener l'acte terroriste. Plutôt que de commettre des actes de violence contre des personnes ou des biens matériels, le cyber-terroriste commet des actes de destruction et de perturbation contre la propriété numérique.»(Denning 2007). La différence entre les hacktivistes et les cyber-terroristes réside dans les spécificités de la cause motivante et de la façon dont nous la qualifions.

Le cyber-espace offre aux groupes terroristes, en plus de la communication, le recrutement, la formation et la propagande, un champ de bataille fertile causant des dégâts physiques importants sur des infrastructures critiques. Il leur assure une action à distance à un coût faible, à un risque minime mais à un impact important.

En janvier 2011, un hacktiviste d'anonimous surnommé « @FuryOfAnon » a dévoilé les logins et les mots de passe des comptes administrateurs de nombreuses infrastructures critiques israéliennes dans le cadre de l'opération « free Palestine ». Entre 2012 et 2013, les hacktivistes du groupe palestinien « Izz ad-Din Al Qassam Cyber Fighter » ont lancé l'opération « ABABIL » visant les intérêts économiques et financiers israéliens et américains causant d'importantes pertes à une trentaine de banques américaines et aux nombreuses entreprises israélo-américaines(Tripathi et al. 2013).

En plus des exemples et des analyses illustrés dans (Denning 2007), citons qu'en 2015, afin d'acquérir des cyber-compétences nécessaires, le groupe terroriste DAISH a lancé, sur le net, une campagne de recrutement des hackers pour préparer des cyber-attaques contre les infrastructures énergétiques et du transport occidentales en leur proposant un salaire de 10.000 Dollars américains par mois.

En décembre 2016, des cyber-terroristes ont réussi à couper le cinquième de l'électricité de la ville de Kiev en piratant le centre de contrôle de la compagnie d'électricité ukrainienne Ukrenergo. Durant plus de 75 minutes, la capitale a vécu un véritable black-out. Une première qui pourrait n'être qu'un galop d'essai. Un an plus tôt, des cyber-terroristes se sont introduits dans les ordinateurs de 3 compagnies régionales d'électricité ukrainiennes puis se sont déplacés physiquement pour éteindre des stations électriques régionales.

4. Crime organisé

Tenir une entreprise en rançon en menaçant d'exploiter les vulnérabilités de leur système SCADA peut sembler une perspective attrayante pour les auteurs des crimes organisés. En effet, les criminels sont susceptibles d'avoir accès à des fonds importants d'où leurs fortes motivations d'embaucher des hackers hautement qualifiés ou d'investir dans le développement et l'acquisition des outils nécessaires pour l'organisation et le lancement des cyber-attaques.

Après le lancement et le test de nombreux rançongiciels entre 2016 et 2017 tels que Peyta, Wannacry, Adylkuzz visant des réseaux d'information, le rançongiciel NotPetya a attaqué, en juin 2017, des systèmes SCADA opérant sous Microsoft Windows en bloquant le lancement des logiciels et des accès aux données sur disques. Le rançonware a causé d'importants dégâts économiques. Il a pu déconnecter le système de surveillance des rayonnements de la centrale nucléaire ukrainienne de Tchernobyl. Il a également infecté les réseaux de métro ukrainiens, la société de publicité britannique WPP, la compagnie pharmaceutique américaine Merck & Co., la compagnie pétrolière russe Rosneft, L'entreprise de construction française Saint-Gobain et ses magasins de détail et filiales en Estonie, la société britannique de biens de consommation Reckitt Benckiser, la société allemande de soins personnels Beiersdorf, la société de logistique allemande DHL, l'opérateur hospitalier américain Heritage Valley Health System, le plus grand port de conteneurs de l'Inde JNPT...(Choi et al. 2016; Fayi 2018; Heaven 2018; Perlroth et al. 2017)

5. Les amateurs et les script-kiddies

Les autres auteurs des cyber-attaques sont les amateurs et les «script kiddies». Les script-kiddies sont souvent définis comme une forme de hackers low life, qui utilisent des scripts gratuits et des outils nécessitant peu de configuration. Les amateurs n'ont généralement pas le financement ou la motivation nécessaire pour acheter des produits coûteux qui peuvent être trouvés sur le marché souterrain. Ces auteurs recherchent souvent un frisson ou un défi pour satisfaire leurs caractères

curieux. Motivés par leur curiosité, ils peuvent causer des dommages importants aux infrastructures ciblées.

En mars 1997, un pirate mineur a pénétré et désactivé un ordinateur de la compagnie de téléphonie qui desservait l'aéroport de Worcester au Massachusetts. Par conséquent, le service téléphonique connectant la tour de contrôle de l'administration de l'aviation fédérale (FAA - Federal Aviation Administration), le service d'incendie et de la sécurité de l'aéroport, du service météorologique et de diverses compagnies privées de fret aérien a été coupé pendant six heures. Plus tard dans la journée, le mineur a désactivé un autre ordinateur de la compagnie causant cette fois ci une panne dans la région de Rutland. La panne a causé des pertes financières et a menacé la santé publique et la sécurité publique(Denning 2000).

En 2008, l'outil de test de pénétration populaire, Metasploit, a publié une nouvelle attaque qui peut affecter les systèmes SCADA Citect (racheter par Schneider Electric) et les rendre défaillants(Nicholson et al. 2012).

En 2009, à Dallas, Etats-Unis, un agent de sécurité de l'hôpital, Jessie William Mc Graw (alias Ghost Exodus), a profité de sa position pour installer des logiciels malveillants sur les machines des hôpitaux en leur permettant de contrôler le système de chauffage, ventilation et climatisation(Nicholson et al. 2012).

6. Les White-hats

Les White-hats sont des hackers qui attaquent les systèmes SCADA afin de tester, détecter et signaler des vulnérabilités aux utilisateurs et/ ou aux constructeurs dans le but de prendre les dispositions et les précautions nécessaires.

En mars 2007, un test, surnommé « Aurora test », a été effectué par le département de l'énergie des états unis (US-DoE, US-Department of Energy) dans le but de déterminer la faisabilité de lancer des cyber-attaques contre les générateurs électriques actuels. Dans ce cadre, Le laboratoire national d'Idaho (Idaho National Laboratory) a pu détruire et rendre hors service un générateur diesel de 3.8 MVA connecté au réseau électrique. Ceci était possible en desynchronisant l'alternateur du générateur par rapport au réseau électrique suite à une cyber-attaque. En pratique, ce style d'attaque peut se produire facilement par une intervention de sabotage sur le système de télécontrôle, sur les relais de protection ou par l'implémentation d'un malware au niveau des programmes de

contrôle(Zeller 2011). Cette expérience a motivé le développement de nouvelles générations de générateurs intelligents qui s'adaptent mieux à la modification brusque des caractéristiques du réseau électrique (Liu et al. 2012; Salmon et al. 2009; Shakarian et al. 2013; Weiss 2016).

7.Hackers non identifiés

Une grande partie des cyber-attaques ciblant les systèmes restent à ce jour sans revendication. En vérité, l'identification de la source et de l'objectif réel d'une cyber-attaque reste un grand défi. En effet, la connectivité des systèmes permet, techniquement, le lancement des attaques programmées et synchronisées de différents terminaux à travers le monde sans que leurs propriétaires ne l'aperçoivent.

En Mai 2001, des hackers, probablement Chinois, ont pu accéder à l'un des réseaux informatiques du (Cal-ISO California Independent System Operator) contrôlant un certain nombre de réseaux SCADA. L'ampleur de ce piratage, qui a duré plus de deux semaines, n'est pas identifié à ce jour(Mustard 2005).

Dans la même année, Le ver industriel SOBIG a interrompu le système SCADA qui gère la signalisation des trains en Floride, aux États-Unis. Le ver, qui exploitait des failles sur les systèmes d'exploitation Windows et qui se propageait par courriers électroniques, a bloqué les systèmes de signalisation, d'expédition et autres systèmes de CSXCorporation ; l'un des plus grands fournisseurs de transport aux États-Unis. Bien qu'il n'y ait eu aucun incident majeur causé par ce cas, les trains ont été gravement retardés(Nicholson et al. 2012).

En effet, en janvier 2003, le ver SQL-Slammer a infecté la centrale nucléaire de Davis Besse dans l'Ohio, aux États-Unis. En raison de l'activité du ver, le système d'affichage des paramètres de sécurité et l'ordinateur de processus de l'usine ont été désactivés pendant plusieurs heures(Miller and Rowe 2012).

En août 2006, la centrale nucléaire de Browns Ferry à Alabama, aux États-Unis, a été fermée manuellement en raison de la défaillance d'un certain nombre de pompes de recirculation du réacteur. Il a été constaté ultérieurement que cette défaillance était causée par la surcharge du trafic réseau qui peut être due à une attaque par déni de service (DoS) (Miller and Rowe 2012).

IV. Outils et types des cyber-attaques

Les cyber-agresseurs des systèmes SCADA adaptent leurs moyens d'attaques en fonction des objectifs et des caractéristiques des segments ciblés. En effet, les cyber-attaques ciblant les MTU et les RTU des systèmes SCADA exploitent des vulnérabilités d'origines internes. Contrairement aux attaques ciblant les segments de réseaux de coopérations qui sont généralement de source externe (Erez and Wool 2015; Genge et al. 2015; Goldenberg and Wool 2013; Huitsing et al. 2008; Knowles et al. 2015).

1. Les cyber-attaques visant les MTU et les RTU

Le segment principal du réseau SCADA est installé principalement dans des centres de contrôle sécurisés à accès limités. La connexion avec ce segment stratégique ne peut être effectuée qu'à partir d'une présence physique des employées, une liaison à distance ou via le réseau de coopération de l'entreprise. Le lancement d'une cyber-attaque visant ce segment ne peut être possible sans intrusions internes exploitant des erreurs de conception des stratégies internes de sécurité, du non-respect des consignes, des actes de sabotage et de vandalisme, ou à l'exploitation des connectivités et les accès ouverts qui assurent les communications distantes avec les utilisateurs et les autres systèmes de l'entreprise.

Le segment des équipements de terrain est déployé sur les locaux et les structures techniques lointains. Il possède généralement un accès aux installations de même niveau ou d'un niveau plus faible que celui dédié au centre de contrôle. Similairement au segment précédent, la quasi-totalité des cyber-attaques, qui le ciblent, exploitent des vulnérabilités internes des installations.

La protection du segment principal et du segment des équipements de terrain peut être efficace par une bonne définition et un rigoureux respect de la stratégie de sécurité qui se base sur l'analyse complète des méthodes d'attaques et de vulnérabilité du système. Nous classifions les cyber-attaques visant ces deux segments en trois catégories :**(*i*)** les attaques par exploits des vulnérabilités stratégiques, **(*ii*)** les attaques par contrôle des accès et **(*iii*)** les attaques par malwares.

Les attaques par exploits des vulnérabilités stratégiques se basent sur l'exploitation des failles constatées dans la stratégie sécuritaire de la gestion des systèmes de contrôles industriels. Cette catégorie contient plusieurs vulnérabilités tels que : (1) l'accès physique facile aux étrangers. (2)Le partage des ressources et des connexions des réseaux industriels avec des applications IT non-sûres

et/ou non-autorisées. (3) L'exploitation des périphériques et des supports de stockages entre différents systèmes d'entreprise, machines, applications ou usagers. (4) L'utilisation de mots de passe faibles non personnalisés ou pour de longues périodes. (5) La compromission des administrateurs et des opérateurs en simplifiant le partage ou l'usage d'une partie de leurs privilèges par d'autres personnes non-autorisées. (6) L'intervention à distance sur des supports, des réseaux ou des ports non-sécurisés, (7) l'absence des équipements de protection softwares et hardwares (antivirus, anti-spam, anti-spys, passerelles, pare-feu, commutateurs intelligents...) à jours dédiés aux systèmes industriels...

Les attaques par contrôle des accès sont généralement réalisées lorsque les auteurs des cyber-attaques peuvent avoir le contrôles des accès (1) aux ports de communication, (2) systèmes d'exploitation, (3) applications, (5) interfaces homme machines ou(6) aux données situés aux niveaux des nœuds des systèmes industriels (MTU/ RTU).

Finalement, la catégorie des attaques par malwares utilise des logiciels fonctionnant sur les machines des deux segments tels que : (1) Les virus, (2) les vers, (3) les chevaux de Troie, (4) les ransomware ; (5) et les mises à jours, les versions non compatibles des logiciels...

2 Les cyber-attaques visant le segment de réseau de coopération

Au jour d'aujourd'hui, le segment réseaux de coopération est constitué de réseaux de télécommunication WAN normés utilisant des technologies, des interfaces et des protocoles ouverts. D'un point de vue sécuritaire, l'utilisation de protocoles et systèmes ouverts a rendu les systèmes de contrôle industriels plus vulnérables aux attaques externes. En effet, il est difficile d'empêcher des tiers à se connecter, à recevoir ou à envoyer des messages sur support utilisé par le système industriel lorsque ce dernier est déployé sur une large zone géographique et utilise des technologies de télécommunication ouvertes. Ceci est encore plus difficile lorsque le support utilisé est hertzien ou publiquement partagé. En général, le suivi d'une bonne stratégie sécuritaire ne suffit pas pour sécuriser ce segment contre les cyber-attaques. Notre analyse des cyber-attaques visant le segment réseaux de coopération des réseaux SCADA nous a permis de les classifier en trois catégories : (***i***) la catégorie des attaques passives, (***ii***) la catégorie des attaques de dénis de service et finalement (***iii***) la catégorie des attaques par modification et fabrication.

Les attaques passives ciblent la confidentialité des systèmes. Elles sont appelées ainsi car elles n'affectent aucun changement sur les parties attaquées. Elles se basent sur l'interception des

données échangées entre les différents nœuds du système afin de collecter un maximum d'information sur les infrastructures critiques contrôlées ou de préparer d'autres attaques actives. Ces attaques peuvent être sur trois niveaux : (1) l'espionnage du canal, (2) l'analyse du trafic et (3) le décryptage des trames.

Le premier niveau d'attaques (l'espionnage du canal) est favorable lorsque l'interception des données SCADA peut, elle seule, être suffisante pour acquérir l'ensemble des informations décrivant l'état du système contrôlé. Le second niveau (l'analyse du trafic) offre, en plus des informations sur l'état des infrastructures, une description de ses équipements et de ses softwares SCADA, des caractéristiques techniques du support de communication, de la stratégie sécuritaire du système, et des objectifs de l'entreprise. Le dernier niveau (le décryptage des trames) est utilisé face aux trames cryptées échangées entre les nœuds du système. La détermination de l'algorithme et de la clé de chiffrement est nécessaire pour l'exploitation des données. Dans ce cas, les agresseurs des systèmes doivent exploiter plusieurs techniques et attaques, telles que l'attaque de recherche en force brute et l'attaque par la table en arc-en-ciel, afin de braiser la confidentialité du système.

Les attaques du Déni de Service (DoS - Denial of Service attacks) sont des attaques actives qui visent à perturber les communications et à interrompre la disponibilité du système visé. Nous pouvons classifier les attaques du déni de service, selon la source, en attaques de déni de service simple (SDoS - Simple Denial of Service attacks) lancées à partir d'un seul point physique, et les attaques du déni de service Distribuées (DDoS - Distributed Denial of Service attacks) lancées à partir d'un ensemble de points physiques.

Les attaques DoS peuvent s'effectuer en visant les différentes couches du segment.(1) Les attaques du déni de service d'accès ciblent les interférences physiques de communications.(2) Les attaques du déni de service réseau ciblent à saturer les réseaux de communication par un bombardement massif des canaux de communication par des signaux de bandes passantes importantes ou par un flux important de paquets.(3) Les attaques du déni de service applicatives se basent sur les interrogations multiples et denses des serveurs afin de les saturer et de les rendre défaillants.

Les attaques de modification et de fabrication sont des attaques actives qui visent l'intégrité et l'authenticité des systèmes. Les agresseurs introduisent leurs messages conçus afin d'obtenir plus de données, de transmettre de fausses informations ou de lancer des instructions de contrôle. Parmi ces attaques nous citons : (1) Les attaques Homme au milieu (Man-in-the-Middle Attacks) ; (2) Les

attaques de répudiation (Repudiation Attacks) ; (3) Les attaques de rejeu (Replay Attacks) ; (4) Les attaques de mascarades (masquerade attacks) ; (5) Les attaques de modification (Modification Attacks)

Les attaques de type homme au milieu sont des attaques assez sophistiquées basées sur l'implémentation, en cachette, des scripts et logiciels malveillants entre les nœuds des réseaux. Ces malwares permettent aux agresseurs de se positionner au milieu des réseaux de communication afin d'intercepter les messages inter-changés en réservant la possibilité d'envoyer leurs propres messages selon la volonté et les besoins des attaquants.

Les attaques de répudiation transforment les données du système en données invalides ou trompeuses. En effet, les attaquants peuvent facilement les lancer en modifiant des parties des trames inter-changées. Par conséquent, les nœuds du système vont ignorer tous les messages modifiés qui ne correspondent pas aux formats prédéfinis des messages.

Les attaques de rejeu sont des attaques basées sur la reproduction des trames déjà interceptées. Dans ce cas, les attaquants ne sont pas obligés de connaître l'interprétation exacte des trames capturées, il leur suffit de rediffuser les mêmes trames pour reproduire les mêmes conséquences.

Les attaques de mascarade sont des attaques qui utilisent une identité falsifiée (telle qu'une identité de réseau) pour obtenir un accès non officiel au système. Les attaques de mascarade sont généralement effectuées en utilisant soit des mots de passe volés et des ouvertures de session, en localisant des lacunes dans les programmes, ou en trouvant un moyen de contourner le processus d'authentification.

Les attaques de modification impliquent l'interception, la suppression, l'insertion et la modification des trames d'une manière non autorisée afin de transmettre une information qui est destinée à paraître authentique à l'utilisateur.

3 Modélisation des cyber-attaques visant les systèmes SCADA

En se basant sur les types d'attaques cités précédemment et sur le modèle des cyber-attaques visant les ICS présenté dans (Huang et al. 2009), nous proposons de les modéliser par la Figure II.1 où nous avons représenté les cyber-attaques internes ciblant le segment réseau principal de contrôle et le segment de réseau de terrain ainsi que les trois principales familles des attaques externes visant le segment réseaux de coopération.

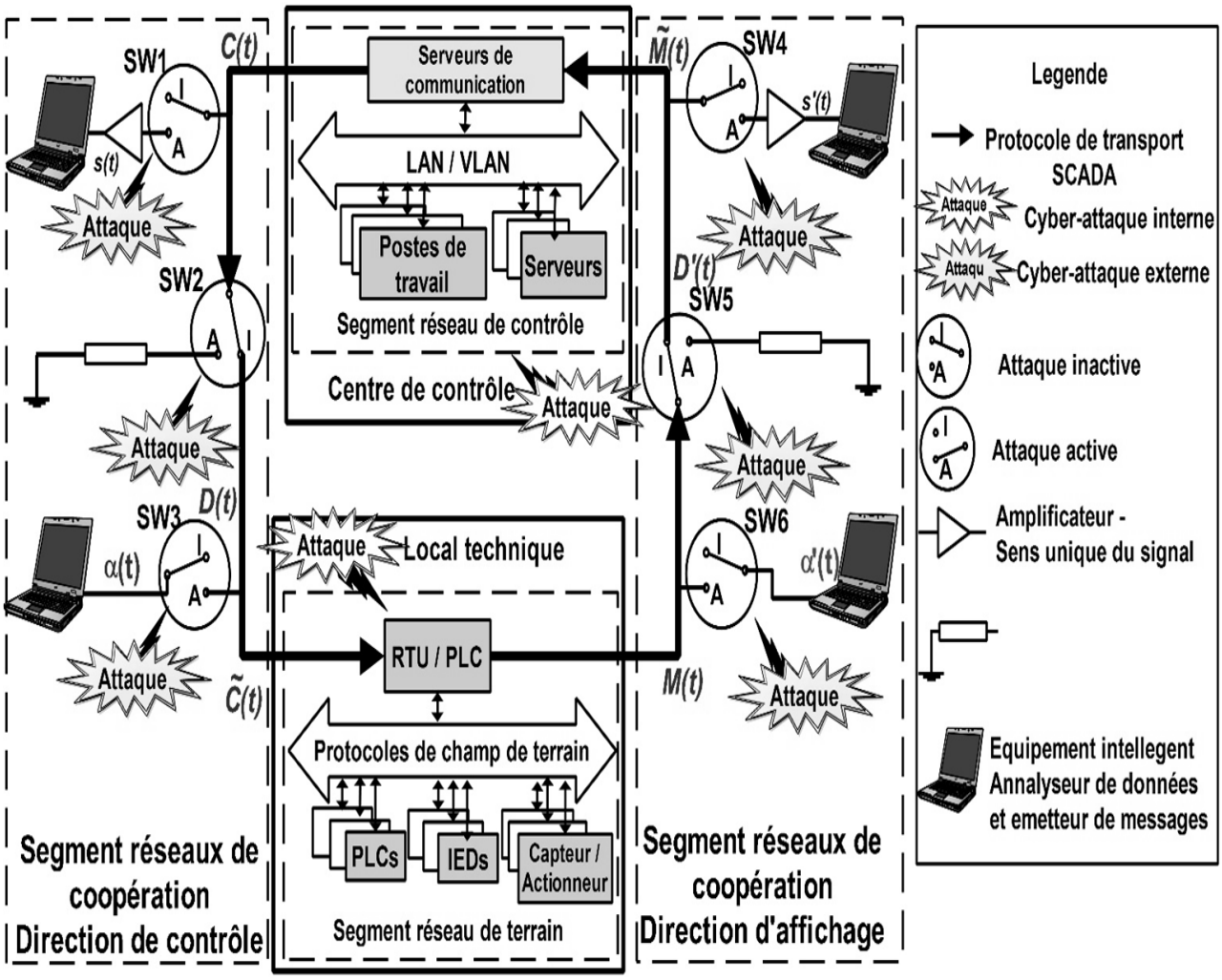


Figure II.1 Modélisation des cyber-attaques visant les systèmes SCADA

Nous avons modélisé les attaques externes par des composants électroniques qui peuvent être alimentés par le signal porteur d'information via des commutateurs SW. Lorsqu'un commutateur SW_ibascule d'une position *I*, de valeur binaire 0, à la position *A*, de valeur binaire 1, l'attaque *i* modélisée est activée et vice versa.

Soit $C(t)$ le signal porteur des trames envoyées par les équipements des MTU dans la direction de contrôle et soit $\tilde{C}(t)$ le signal reçu par les équipements de terrain. De même, soient $M(t)$ et$\tilde{M}(t)$, respectivement, le signal porteur des trames envoyées par les équipements de terrain dans la direction d'affichage et le signal reçu par les équipements de contrôle.

Les commutateurs SW_1 et SW_4 modélisent les contrôleurs des attaques passives. Quand ils passent à la position active, les attaquants seront capables de recevoir les données sans les modifier. Nous

avons choisi de modéliser les attaques passives par des amplificateurs. En effet, les signaux capturés porteurs de trames sont toujours entrants et ils sont, dans la majorité des cas, prétraités afin de compenser la fonction de transfert du canal (amplifier pour compenser l'atténuation...). Nous noterons ces signaux par $s(t)$ lorsqu'ils sont dans la direction de contrôle et par $s'(t)$ lorsqu'ils sont dans la direction d'affichage. Nous pouvons donc écrire :

$$\begin{cases} s(t) = SW_1 \times C(t) \\ s'(t) = SW_4 \times M(t) \end{cases} \quad (2.1)$$

Les commutateurs SW_2 etSW_5 modélisent les contrôleurs des attaques par déni de service. Lorsqu'ils sont sur la position active, le canal de transmission est rompu et toutes les communications entre le segment principal de contrôle et le segment réseaux de terrain sont interrompues. Par conséquent, nous avons choisi de modéliser ces attaques par des circuits de faibles impédances reliés à la terre. Soit $D(t)$ et $D'(t)$ les signaux présents, respectivement, dans le canal de communication après SW_2 et SW_5.

$$\begin{cases} D(t) = \overline{SW_2} \times C(t) \\ D'(t) = \overline{SW_5} \times M(t) \end{cases} \quad (2.2)$$

Les commutateurs SW_3 et SW_6 modélisent les contrôleurs des attaques par modification et fabrication. Lorsqu'ils sont actifs, les agresseurs seront capables de modifier les signaux présents dans le canal et de lancer leurs propres signaux. Soit $\alpha(t)$ et $\alpha'(t)$ les deux fonctions de modification définies respectivement dans le sens du contrôle et dans le sens d'affichage

$$\begin{cases} \tilde{C}(t) = D(t) + \left(SW_3 \times \alpha(t)\right) \\ \tilde{M}(t) = D'(t) + \left(SW_3 \times \alpha'(t)\right) \end{cases} \quad (2.3)$$

En remplaçant le système d'équation (2.2) dans les équations du système (2.3), nous allons trouver :

$$\begin{cases} \tilde{C}(t) = \left(\overline{SW_2} \times C(t)\right) + \left(SW_3 \times \alpha(t)\right) \\ \tilde{M}(t) = \left(\overline{SW_5} \times M(t)\right) + \left(SW_3 \times \alpha'(t)\right) \end{cases} \quad (2.4)$$

Nous pouvons remarquer que ce modèle exprime, clairement, que les attaques ciblant le segment des réseaux de coopération peuvent défier les principaux objectifs sécuritaires. En effet, les relations de (2.1) modélisent les attaques passives ciblant la confidentialité des données. Les équations (2.2)

modélisent les attaques sur la disponibilité du système, alors que les relations (2.3) modélisent les attaques ciblant l'authenticité et l'intégrité des messages inter-changés.

Les relations (2.1) et (2.3) traduisent, clairement, l'indépendance entre les messages capturés par les attaques passives ($s(t)$ et $s'(t)$)d'une part et les messages reçus par les équipements terminaux du système SCADA $\left(\tilde{C}(t) \text{ et } \widetilde{M}(t)\right)$ d'une autre part. Cette indépendance exprime la difficulté de la détection de ces attaques silencieuses qui s'adressent à la confidentialité des messages inter-changés.

Certains spécialistes de la sécurité des systèmes SCADA considèrent la confidentialité comme un objectif sécuritaire moins important que la disponibilité, l'authenticité et l'intégrité des données(Wei et al. 2010). Pour notre part, nous nous opposons à cet avis pour deux raisons : Premièrement, les données SCADA inter-changées contiennent des informations instantanées sur les infrastructures contrôlées tels que son importance, son état, ses activités, ses faiblesses... Ces informations peuvent être exploitées pour déterminer le meilleur moment ainsi que la meilleure technique pour lancer une attaque physique ou une cyber-attaque active. D'un autre point de vue, la confidentialité peut aider à maintenir les autres objectifs sécuritaires, notamment l'authenticité. En effet, les membres du système peuvent encrypter les messages envoyés et décrypter les messages reçus correctement. Lorsqu'un message décrypté présente des erreurs dans sa structure (comme les inversions dans l'ordre des octets) ou dans sa valeur (comme la détection d'une valeur inacceptable d'un octet), il doit être totalement ignoré. Ceci offre l'exclusivité aux membres du système, procédant les clés du chiffrement, de communiquer entre eux(Cherifi and Hamami 2018).

V. Normes et stratégies de cyber-protection des systèmes SCADA

Dans le but de la sécurisation des systèmes SCADA, plus de 40 normes régionales, nationales ou internationales, contenant des recommandations, des lignes directrices et de bonnes pratiques ont été définies. Nous citons dans le Tableau II-1 les 15 standards qualifiés d'être actualisés, spécialisés en ICS, utilisés dans de larges zones géographiques, et définis par des organisations internationales ou des nations fortement impliquées dans la cyber-guerre moderne (Zhou et al. 2017).

Tableau II-1 Les 15 Standards sélectionnés de protection des systèmes SCADA

Standard	Organisation	Nation	Année	Conception de système	Fabricants des équipements	propriétaire	Certification et évaluation
Security for Industrial Automation and Control Systems (**IEC 62443/ ISA 99**)	International Electrotechnical Commission (**IEC**), International Society of Automation (**ISA**)	Inter	2007	+	+	+	+
Security in power system control operations (**IEC 62351**)	International Electrotechnical Commission (**IEC**)	Inter	2018	+	+	+	+
Critical infrastructure Protection **(CIP)-002 – CIP-011**	North American Electric Reliability Council-**NERC**	Rég	2015	+	+	+	+
Guide to industrial control systems (ICS) security (**SP800-82**)	National Institute of Standards & Technology-**NIST**	USA	2015	+	+	+	+
System Protection ProfileCIndustrial Control Systems (**NISTIR-7176**)			2004	-	-	+	+
Guidelines for Smart Grid Cyber Security (**NISTIR-7628**)			2010	+	+	+	+
Cyber security procurement language for control systems (**DHS-LANGUAGE**)	Department of Homeland Security **U.S.DHS**		2017	+	-	-	+
Strategic principles for securing the internet of things (**DHS-STRATEGIC**)			2016	+	+	+	+
21 steps to improve cyber security of SCADA networks (**DOE-21STEPS**)	U.S.Department of Energy **U.S.DOE**		2002	-	-	+	+
Cyber security for critical infrastructure protection (**GAO-Protection**)	U.S.Government Accountability Office (**U.S.GAO**)		2004	+	+	+	+
Good Practice Guide C Process Control and SCADA Security (**CPNI PRACTICE**)	Center for the Protection of National Infrastructure (**U.K.CPNI**)	UK	2005	+	+	+	+
Industrial control system Security Part 1-2 (**GB/T30976.1-2-2014**)	Standardization Administration of the People's Republic of China (**P.R.SAC**)	RP-Chine	2014	+	+	+	+
Evaluation specification for security in industrial control network (**GB/T26333-2010**)		RP-Chine	2010	-	-	+	+
Industrial-process measurement and control–Evaluation of system properties for the purpose of system assessment (**GB/T18272.1-8-2000**)		RP-Chine	2000	-	-	+	+
Guide to industrial control systems information security protection (**MIIT GUIDE**)	Ministry of Industry and Information Technology **P.R.MIIT**	RP-Chine	2016	+	+	+	+

L'IEC-62443, connu aussi par ISA99 ou (IACS - Industrial Automation and Control Systems Security) est la série de normes internationales la plus représentatives et la plus discutée. Conçue et publiée par IEC en 2007, elle regroupe plus de 12 documents organisés en 4 sections. L'IEC 62443-1-1 introduit les concepts, les modèles et la terminologie utilisés par le standard. L'IEC 62443-1-2 fournit le glossaire principal des termes et la liste des abréviations trouvées dans les documents. L'IEC 62443-1-3 utilise les exigences fondamentales, les exigences des systèmes et d'autres informations pour tester les spécificités des métriques sécuritaires quantitatives. L'IEC 62443-2-1 décrit les exigences de la définition et l'implémentation des systèmes de management de la cyber-sécurité des IACS. L'IEC 62443-2-2 décrit les exigences du fonctionnement des systèmes de management de la cyber-sécurité des IACS. L'IEC 62443-2-3 présente des recommandations aux propriétaires des installations contrôlées et aux développeurs de solutions lors de l'établissement des programmes de gestion de la sécurité lors de la maintenance des IACS. L'IEC 62443-2-4 spécifie les exigences de la capacité sécuritaire des fournisseurs de services IACS offerte aux propriétaires des installations lors de l'intégration et la maintenance des activités des solutions automatisées. L'IEC 62443-3-1 est un rapport technique qui propose l'application de plusieurs technologies de sécurité dans l'environnement IACS. L'IEC 62443-3-2 décompose les IACS en segments. L'IEC 62443-3-3 présente les exigences sécuritaires des systèmes et les niveaux d'assurances sécuritaires. L'IEC 62443-4-1 définit les exigences sécuritaires lors du développement des solutions IACS. Finalement, l'IEC 62443-4-2 présente un ensemble de recommandations dérivées illustrant des détails sur les sous-systèmes et les composants du système sécurisé(Byres et al. 2012; Piggin 2013; Zhou et al. 2017).

L'IEC 62351 est une norme récente développée pour la sécurité des opérations de contrôle des systèmes électriques. La norme IEC 62351 est une norme composée de 11 parties pour couvrir tous les aspects de la sécurité dans la communication des services d'électricité. Les parties 1 et 2 présentent les spécifications techniques traitant des problèmes de sécurité dans les systèmes de contrôle de puissance. Les parties 4, 5, 6 sont publiées en tant que spécifications techniques sur la façon d'implémenter la sécurité dans les protocoles de communication de contrôle de puissance, tels que MMS, IEC61850 sur TCP / IP. En plus de la norme de sécurité pour les protocoles de communication, les parties 7 à 11 de l'IEC 62351 couvrent la sécurité de bout en bout, impliquant des politiques de sécurité, des mécanismes de contrôle d'accès, la gestion des clés, le journal d'audit et d'autres problèmes de protection des infrastructures critiques.(Cleveland 2012; Drias et al. 2015)

La série des documents CIP-002-CIP-011 a été publiée pour la première fois en 2005 puis complétée le 13 février 2015 par la NERC pour protéger les infrastructures critiques et leurs systèmes ICS. Ses documents s'intitulent : (CIP-002) Identification du cyber-actif critique, (CIP-003) Contrôles de la gestion de la sécurité, (CIP-004) Personnel et formation, (CIP-005) Périmètre de sécurité électronique, (CIP-006) Sécurité matérielle des cyber-actifs critiques, (CIP-007) Gestion de la sécurité des systèmes, (CIP-008) Rapports d'incidents et planification des interventions, (CIP-009) Plans de rétablissement pour les cyber-actifs critiques, (CIP-010) Gestion des changements de configuration et évaluation de la vulnérabilité, et finalement (CIP-011) Protection de l'information(Bui et al. 2016).

Le SP800-82 est une actualisation du guide des systèmes SCADA publié par NIST en 2004 (Stouffer et al. 2006). Il fournit des recommandations pour sécuriser tous les types de systèmes de contrôle industriel ICS, tout en répondant à leurs exigences uniques de performance, de fiabilité et de sécurité. Ce guide fournit une vue d'ensemble des ICS en définissant les topologies des systèmes, en identifiant les menaces et les vulnérabilités et en fournissant des contre-mesures de sécurité recommandées pour atténuer les risques associés(Stouffer et al. 2011).

Le NISTIR7176 est une norme développée par NIST en 2004 pour définir le profil de protection des systèmes industriels (SPP – System Protection Profile). Ce document est organisé en 8 sections où la première section fournit le matériel d'introduction pour le profil de protection du système. La deuxième section présente le but général et une description des systèmes cibles d'évaluation (STOE-System Target of Evaluation). La troisième section offre une discussion de l'environnement attendu pour le STOE. Cette section définit également l'ensemble des menaces qui doivent être traitées par les techniques opérationnelles de contrôle et de gestion mis en œuvre par le STOE. La quatrième section identifie les risques dérivés de l'énoncé de l'environnement de sécurité défini à la section 3. La cinquième section définit les objectifs de sécurité pour les environnements et STOE. La sixième section contient les exigences fonctionnelles et d'assurance qui doivent être respectées par le STOE. La septième section contient des orientations pour les concepteurs des systèmes de sécurité qui souhaitent revendiquer la conformité au SPP. La dernière section explique la relation entre le STOE, les risques et les objectifs sécuritaires (Melton et al. 2004).

Le NISTIR-7628 a été publié pour la première fois en août 2010. Le contenu du rapport comporte 3 volumes. Le volume 1 présente la stratégie, l'architecture et les exigences de haut niveau en matière

de cyber-sécurité. Le volume 2 porte sur la difficulté de protéger la confidentialité des Smart Grids. Le volume 3 s'intéresse à la classification de vulnérabilité, l'analyse de la sécurité ascendante des Smart Grids et les thèmes de recherche et développement(Brooks et al. 2015).

Le DHS-LANGUAGE a été publié en août 2006. La version la plus récente est 1.8, révisée en février 2017. Il contient des recommandations sur les critères de choix des systèmes matériels de protection périmétrique, la gestion des comptes, les pratiques de codage, les techniques de correction des erreurs, les méthodes de détection et protection contre les logiciels malveillants, la résolution des noms d'hôtes et de terminaux, les accès à distance, la sécurité physique et le partitionnement des réseaux (Finco et al. 2007).

La dernière version du DHSSTRATEGIC est publiée le 15 novembre 2016, 26 jours après l'attaque massive DDoS bloquant l'accès aux sites Web américains. Ses principes soulignent les approches et les pratiques suggérées pour renforcer la sécurité des objets connectés par internet IoT. Elles sont basées sur les domaines clés traitant l'intégration de la sécurité à la phase de conception des systèmes ; le déroulement des mises à jour des systèmes de sécurité ; la gestion des vulnérabilités ; l'appui sur des pratiques de sécurité éprouvées et la promotion de la transparence dans l'ensemble de l'écosystème IoT...(Kraft and Marks 2016; Tama 2016)

Le DoE 21STEPS a été publié en 2002. Le Presidents Critical Infrastructure Protection Board et le ministère de l'Énergie américains ont élaboré 21 étapes pour aider toute organisation à améliorer la sécurité de ses réseaux SCADA. Ces étapes ne sont pas censées être normatives ou exhaustives. Cependant, elles traitent des actions importantes pour améliorer la protection des réseaux SCADA. Elles sont divisées en deux catégories : des actions spécifiques pour améliorer les implémentations des systèmes et des actions pour établir des processus et des politiques de gestion(Energy 2005; Sommestad et al. 2010).

Le GAO Protection est un standard américain publié en 2004. Il propose d'utiliser des technologies de cyber-sécurité pour la protection des infrastructures critiques. Il contient principalement trois parties. La première présente les exigences cyber-sécuritaires des différents secteurs d'infrastructures critiques. La deuxième illustre les technologies et les normes de la cyber-sécurité. La dernière partie traite les questions liées à l'implémentation des mécanismes de la cyber-sécurité(Dacey 2004).

En 2005, le U.K.CPNI a publié son standard CPNI-PRACTICE qui regroupe des recommandations pour la sécurisation de chaque niveau des systèmes de contrôle industriel commençant par les processus de contrôle, les automates, les DCS, les systèmes SCADA et définit les outils de protection contre les attaques électroniques. Il vise principalement à sécuriser les systèmes SCADA, illustrer les risques commerciaux liés à la cyber-sécurité, implémenter une architecture sécurisée avec de grandes capacités d'intervention, et de bonnes compétences de gestion des risques(Piggin 2012).

Composé de deux parties, le standard chinois GB / T 30976.1-2-2014 a été publié en 2014. Sa première partie GB / T 30976.1 est destinée aux concepteurs de systèmes, fabricants d'équipements, intégrateurs de solutions, sociétés d'ingénierie, utilisateurs, propriétaires d'actifs et organismes d'évaluation et de certification pour évaluer la sécurité de l'information des systèmes de contrôle industriels. Elle spécifie les objectifs de l'évaluation de la sécurité de l'information des systèmes de contrôle industriels... la deuxième partie GB / T30976.2 spécifie le processus, le contenu du test, la méthode et les exigences de l'acceptation de la sécurité de la solution. Les utilisateurs de cette partie peuvent ajouter des équipements ou des systèmes pour améliorer la sécurité des IACS. Le contenu de cette partie peut être utilisé comme guide dans les travaux pratiques applicables aux industries pétrolières, chimiques, électriques, nucléaires, transport, métallurgie, traitement de l'eau, fabrication et autres(Hao et al. 2016; Trappey et al. 2017).

Le standard chinois GB / T 26333-2010, publié en 2010, présente des méthodes d'évaluation des risques de sécurité pour les réseaux de contrôle industriels. Grâce à cette évaluation de ces risques, nous pouvons estimer les dangers cachés du réseau et combler les failles de sécurité en utilisant les mesures de sécurité correspondantes.

Le GB / T 18272.1-8-2000, publié en 2000, est un autre document chinois qui s'intéresse à l'évaluation de la sécurité des systèmes industriels. Il est composé des parties suivantes :(1) Généralités et méthodologie, (2) Méthodologie d'évaluation, (3) Évaluation fonctionnelle du système, (4) Évaluation de la crédibilité du système, (5) Évaluation opérationnelle du système, (6) Évaluation de la sécurité du système et (7) Évaluation indépendante des caractéristiques du système.

Le ministère chinois de l'information et des technologies de l'information a publié en octobre 2016 le MIIT-GUIDE. Ce guide est destiné aux entreprises et aux institutions chargées de la planification, de la conception, de la construction, de l'exploitation, de la maintenance et de l'évaluation des

systèmes de contrôle industriel. Il contient des lignes directrices pour l'amélioration du niveau de sécurité des informations du contrôle industriel en intervenant sur les critères de la sélection de logiciels, la configuration et la gestion des correctifs, la protection physique, l'authentification, la sécurité d'accès à distance, la surveillance et la réponse, la sécurité des données, la gestion de la chaîne de sécurité des données et la définition des responsabilités (MIIT 2016; Trappey et al. 2017).

D'un point de vue plus global, les premières stratégies, nationales, régionales et internationales, de la cyber-sécurité sont apparues entre 2009 et 2013 dans plusieurs pays du monde. Une stratégie nationale de cyber-sécurité est une stratégie documentée qui illustre le plan d'action national visant à achever un ensemble d'objectifs sécuritaires. Elle doit définir la cyber-sécurité et le cyber espace à protéger, les menaces et les vulnérabilités de l'espace, la vision, les objectifs, les normes et les principes sécuritaires, les textes légaux, les structures et les organismes de contrôle et de protection acteurs dans la cyber-sécurité ainsi que les différentes relations entre la stratégie nationale de la cyber-sécurité et les autres stratégies nationales. L'analyse comparative des stratégies de la cyber-sécurité nationales de 19 pays, à savoir L'Australie, le Canada, la Tchécoslovaquie, l'Estonie, la France, l'Allemagne, l'Inde, le Japon, la Lituanie, le Luxembourg, la Roumanie, les Pays Bas, la Nouvelle Zélande, l'Afrique du sud, l'Espagne, l'Uganda, le Royaume Uni, les Etats Unis et la Russie (Luiijf et al. 2013; Luiijf et al. 2011) nous a montré que les aspects sécuritaires des réseaux de contrôles industriels et des réseaux IT ont été confondus dans la majorité des cas. De ce fait, aucune stratégie n'a pris en compte les spécificités des segments réseaux de coopération utilisés par les réseaux de contrôles industriels.

VI Conclusion

Aujourd'hui, les systèmes de contrôle industriels sont devenus indispensables pour la gestion des infrastructures critiques, les employés mécontents, les pirates étatiques, les hacktivistes, les cyber-terroristes, les cyber-criminels, les amateurs de l'informatique et les white-hats ont trouvé dans les systèmes SCADA un champ de bataille fertile pour lancer leurs cyber-attaques. Motivés par des convictions nationalistes, politiques, religieuses, idéologiques, matérielles, scientifiques, sociales ou environnementales, ces auteurs exploitent tous leurs compétences en informatique classiques et leurs connaissances en systèmes de communication ouverts pour cibler les différents segments des systèmes SCADA. Généralement, le segment principal et le segment champ de terrain sont conçus à partir de réseaux LAN. Ils sont plus vulnérables aux attaques de sources internes qu'aux attaques

externes. Contrairement au segment réseaux de coopération qui est plus vulnérable aux attaques externes.

Afin de se protéger face à ces menaces, plusieurs standards ont été élaborés, l'analyse des 15 standards les plus répandus nous a montré qu'aucun d'entre eux n'a proposé des méthodes pratiques pour sécuriser le segment réseaux de coopération. Certains standards et travaux de recherche, tels que (Ďuďák et al. 2016; Shahzad et al. 2015), ont proposé d'utiliser des crypto-systèmes adaptés aux systèmes IT pour sécuriser les protocoles de transport SCADA. Ces crypto-systèmes semblent inadéquats aux spécificités des systèmes SCADA. Nous proposons de développer, dans le prochain chapitre, un nouveau crypto-système hautement sécurisé permettant de conserver le bon fonctionnement de ces réseaux industriels.

CHAPITRE III : NOUVEAU CRYPTO-SYSTEME POUR LES SYSTEMES SCADA

I. Introduction

Depuis le début des civilisations, le besoin de cacher, de dissimuler des informations personnelles ou confidentielles préoccupe l'humanité, et cela bien avant l'ère informatique, mais avec l'évolution des nouveaux systèmes d'information et de communication la nécessité de protéger le contenu de certains messages d'ordre personnel ou commun s'est imposée. La confidentialité apparaissait notamment nécessaire lors des luttes pour l'accès au pouvoir puis devient énormément développée pour des besoins militaires et diplomatiques.

Aujourd'hui, de plus en plus d'applications dites civiles nécessitent la sécurisation de leurs données, transitant entre les interlocuteurs via les vecteurs d'information actuels qualifiés d'être publiquement accessibles : Les réseaux de télécommunication actuels, les banques, les hôpitaux, les entreprises économiques, les laboratoires de recherche, les associations, les responsables politiques et militaires comme les simples citoyens possèdent tous des informations confidentielles à protéger. La cryptographie est, généralement, définie comme l'art et la science de maintenir la confidentialité des données cryptées pour les rendre intelligibles (Schneier 1994). Elle regroupe l'ensemble des méthodes qui permettent de chiffrer un texte clair et de le rendre incompréhensible sans la possession de la clé de décryptage. Avec l'évolution des systèmes d'information et de communication modernes, le développement et la standardisation de nouveaux crypto-systèmes modernes pour des applications grand public sont devenus incontournables. Après la deuxième guerre mondiale, plusieurs travaux ont été menés en ce sens.

Du point de vue sécuritaire, les systèmes SCADA sont très différents des systèmes IT. Afin de nous permettre le développement d'un protocole SCADA sécurisé, face aux cyber-attaques passives et/ou cyber-attaques par fabrication et modification, nous avons consacré ce chapitre à l'analyse et

l'exploitation des propriétés des crypto-systèmes adaptés aux systèmes IT afin de définir les règles à suivre pour la conception d'un protocole SCADA hautement sécurisé.

II. Principes de la cryptographie moderne

Après la deuxième guerre mondiale, C. E Shannon a dirigé ses travaux dans les laboratoires Bell pour devenir l'un des leaders en Théorie de l'information (codage de l'information, analyse et compression des données, cryptage et cryptanalyse...). Selon le modèle proposé par C. E. Shannon (Shannon 1949), un système cryptographique, dit aussi crypto-système, est basé, mathématiquement, sur deux fonctions, une fonction d'encryptage (E) et sa fonction inverse dite fonction de décryptage (D). La fonction d'encryptage est une fonction injective connue, utilisée pour transformer des messages clairs *M*, produits par une source de message S_m, en messages cryptés *C* par l'utilisation d'une clé d'encryptage K_E, générée par une source de clés S_K. Nous notons l'ensemble des messages cryptés *C* par S_C

$$E\ (M,K_E)=C \tag{3.1}$$

A la réception des messages, la fonction de décryptage *D* est utilisée pour calculer les messages clairs à partir des messages cryptés *C* et la clé de décryptage K_D de la source des clés

$$D(C,K_D)=M \tag{3.2}$$

La Figure III.1 présente le diagramme d'un crypto-système selon le modèle de Shannon.

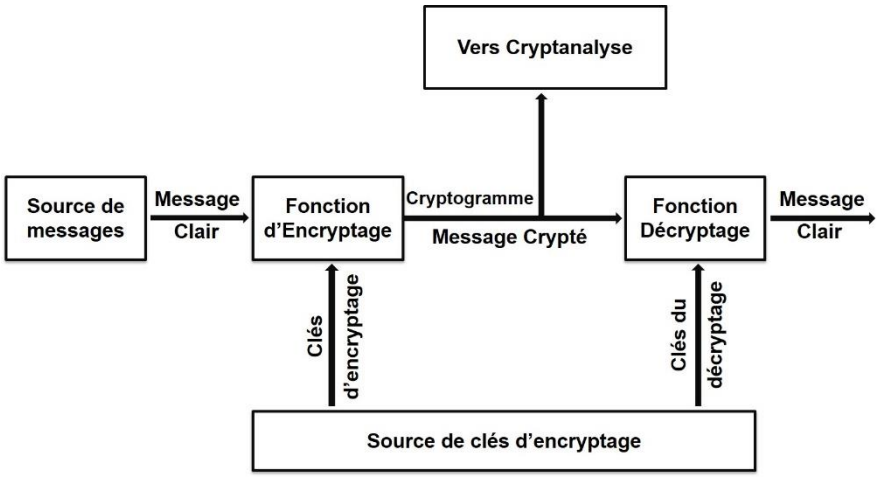


Figure III.1 Diagramme d'un crypto-système

Malgré l'ancienneté de la cryptographie, ce n'est qu'en 1883 que A.KERCKHOFF a etabli le principe de la cryptographie moderne basée sur ses six desiderata à savoir: **(1)** Le système doit être

matériellement, sinon mathématiquement indéchiffrable ; **(2)** Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ; **(3)** La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ; **(4)** Il faut qu'il soit applicable à la correspondance télégraphique ; **(5)** Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ; **(6)** Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer(Petitcolas 2011).

Ceci signifie que la confidentalité de l'information échangée ne doit pas être protégée par un processus secret. Mais elle doit être assurée par une clé de cryptage secrète. En d'autres mots, même si un cryptanalyste pourrait recevoir les messages cryptés et connaitre les fonctions d'encryptage et de décryptage du cryptosystème en ignorant les clés de décryptage, il restera incapable de reconstruire les messages clairs avec certitude. Nous pouvons donc écrire :

$$P(M_i/E, D, C_i) \neq 1 \tag{3.3}$$

Ou :

$$H(M_i/E, D, C_i) \neq 0 \tag{3.4}$$

Où P et H sont respectivement la fonction de probabilité et la fonction de quantité d'information conditionnelles.

III. Classification des algorithmes de cryptage

En se basant sur le modèle de Shannon présenté dans la Figure III.1, nous pouvons distinguer deux familles de crypto-systèmes : les cryptosystèmes symétriques et les crypto-systèmes asymétriques. Un cryptosystème est dit symétrique si sa clé d'encryptage est identique à sa clé de décryptage. Dans le cas opposé, le système est asymétrique (Menezes et al. 1996).

1. Algorithmes de cryptage symétrique ou à clé privée

Pour un crypto-système symétrique, dit aussi un crypto-système à clé privée, les algorithmes de cryptage utilisent des clés d'encryptage et de décryptage identiques (K_E=K_D=K) et partagées entre tous les tiers communiquant et d'eux seuls. Ces crypto-systèmes sont généralement conçus à partir d'une série de transpositions et de substitutions des mots de messages clairs en fonction de ces clés,

et acceptent de fonctionner selon l'un des deux modes : mode de cryptage par blocs (block cipher) ou le mode de cryptage par flots continus (stream cipher) (Menezes et al. 1996).

Un cryptage symétrique par blocs opère sur les messages clairs en les décomposant en larges blocs de mots. Ce mode est plus adapté au cryptage de fichier avant la transmission. Par contre, un cryptage symétrique par flots continus traite chaque mot indépendamment des autres, il est plus adapté au cryptage des canaux de communication vue la simplicité et la rapidité des algorithmes exploités basés sur le principe de l'utilisation unique des clés de cryptage. En effet, la réutilisation des clés de cryptage par flots affaiblit la sécurité du système. En plus de cet inconvénient, l'utilisation de cryptage par flots continus nécessite une parfaite synchronisation entre les différents utilisateurs du système.

L'utilisation des algorithmes de cryptage à clé privée présente plusieurs avantages notamment sa rapidité et sa capacité de crypter de grandes quantités de données. Néanmoins, le partage et le maintien du secret de la clé présente un grand défi. Pour une meilleure sécurité, on préfère utiliser une clé différente pour chaque couple d'utilisateur avec un partage manuel de clés. Ceci semble être difficile, voire impossible pour des systèmes de tailles importantes. En effet, pour N communicants, nous avons besoin de créer et de partager $N.(N-1)/2$ clés(Menezes et al. 1996).

2. Algorithmes de cryptages asymétriques ou par clé publique

Les algorithmes asymétriques sont relativement nouveaux par rapport aux algorithmes symétriques. Ils ont été développés, dans les années 1970, pour répondre au besoin d'échanges sécurisés d'information en présence d'attaques passives et de risque de perte de la confidentialité des clés. Dans les algorithmes asymétriques, les clés d'encryptage et de décryptage sont distinctes et ne peuvent se déduire l'une de l'autre. On peut donc rendre l'une des deux clés publique et l'autre privée.

Le principe de ce cryptage se base, généralement, sur l'exploitation des fonctions à sens unique avec une brèche secrète :

Lorsqu'une entité (A) veut échanger des messages avec une autre entité (B), elle développe une fonction à sens unique avec une brèche secrète basée sur une clé secrète connue uniquement par (A). Cette fonction est tramnsmise à l'entité (B), comme clé publique, pour qu'elle soit utilisée lors de

l'encryptage. Dans ce cas, même si tous les connectés au canal intercepte les messages cryptés, personne d'autre que (A) ne pourra le décrypter.

Ces algorithmes représentent une révolution technolgique dans le domaine de la sécurité d'information. Aucun besoin de distribuer des clés secrètes sur les intervenants, chaque intervenant peut construire sa propre fonction à sens unique et de l'utiliser comme une clé publique pour échanger les données. Par conséquent, le nombre de clés utilisées est réduit au nombre de communiquants. Cependant, ces algorithmes sont basés sur des calculs mathématiques complexes, leur inconvénient majeur , alors, réside dans leur vitesse, environ 1000 fois plus lente que les systèmes à clé privée(Menezes et al. 1996).

IV. Classification des cyber-attaques cryptanalytiques

La cryptanalyse est une discipline de la cryptologie qui s'intéresse à l'étude des faiblesses des crypto-systèmes et au développement de méthodes permettant de briser la protection et de déterminer des messages clairs à partir des messages cryptés en absence d'information a priori sur les clés de cryptage.

Une cyber-attaque est une tentative de cryptanalyse lancée contre un crypto-système pour obtenir, des informations cachées, les messages cryptés.

1. Classification des cyber-attaques selon les objectifs

Le rôle principal d'une cyber-attaque est de briser le dispositif sécuritaire protégeant les informations cachées dans les données cryptées. De ce fait, nous identifions quatre niveaux de rupture du cryptage (Knudsen 1994) : (1) La rupture totale (Total break) qui permet à l'attaquant de retrouver tous les secrets du crypto-système par la reconnaissance des clés secrètes K_E et K_D du cryptage. (2) La déduction globale qui permet au cryptanalyste de déterminer une fonction A indépendante des clés du cryptage pour remplacer la fonction d'encryptage E ou la fonction de décryptage D. (3) La déduction locale où le cyber-attaquant peut définir un ensemble de couples composés des messages clairs avec leurs messages cryptés correspondants. Finalement (4) la déduction d'information qui permet aux cyber-analystes d'acquérir une information supplémentaire sur les caractéristiques du crypto-système.

2. Classification des cyber-attaques selon l'accessibilité aux données

Selon le principe de KERCKHOFF, nous supposons que le cryptanalyste peut intercepter tous les messages cryptés et connaître tous les algorithmes, les fonctions mathématiques et les protocoles du cryptage exploité. Même dans le cas opposé, nous ne devons pas nous baser sur le secret du mécanisme choisi pour sécuriser le système. En effet, il est toujours possible de remonter au crypto-système utilisé en analysant une quantité suffisante de messages échangés. L. R. Knudsen a classifié ces attaques cryptographiques selon les sources cryptanalytiques en cinq principaux niveaux(Jerman-Blažič et al. 2001; Knudsen 1994) à savoir (1) Les attaques à texte chiffré seulement (ciphertext only attacks), (2) les attaques à texte clair connu (known-plaintext attacks), (3) l'attaque à texte clair choisi (chosen-plaintext attack), et (4) les attaques à texte chiffré choisi (chosen-ciphertext attacks).

Les attaques à texte chiffré seulement visent à déterminer les clés du cryptage ou/ et les messages clairs en se basant uniquement sur un ensemble de messages cryptés capturés. Autrement dit, l'ennemi cryptanalyste n'a aucune information additionnelle autre que les messages cryptés capturés. Autrement dit :

$$T \subseteq S_C : H(S_K / E, D, T) = 0 \qquad (3.5)$$

Dans le cas de l'attaque à texte clair connu (known-plaintext attack), le cryptanalyste essaye de déterminer les clés de cryptage en se basant sur un ensemble de couples de messages clairs et leurs messages cryptés. Ceci peut se traduire par :

$$\exists\, L \subseteq S_M : H\big(S_K / \big(E, D, L, E(L)\big)\big) = 0 \qquad (3.6)$$

Contrairement aux attaques à texte choisi, les attaques à texte clair choisi permettent aux cryptanalystes de choisir, a priori, des messages clairs et de connaître leurs messages cryptés puis de les utiliser pour déterminer les clés du cryptage

$$L \subseteq S_M : H\big(S_K / \big(E, D, L, E(L)\big)\big) = 0 \qquad (3.7)$$

Les attaques à texte clair choisi adaptatives (Adaptative-chosen-plaintext attacks) est un cas particulier de l'attaque à texte clair choisi, les cryptanalyses peuvent connaître, en plus des messages cryptés des messages clairs choisis, la correspondance entre les différents blocs des messages clairs et cryptés.

Finalement dans le cas des attaques à texte chiffré choisi, le cryptanalyste possède la possibilité de choisir, a priori, des messages cryptés et de connaître leurs messages clairs puis de les utiliser pour déterminer les clés du cryptage :

$$T \subset S_C : H\left(S_K / \left(E, D, T, D(T)\right)\right) = 0 \qquad (3.8)$$

Il est intéressant de noter que les attaques à textes choisis sont les attaques les plus fortes, mêmes si elles sont généralement difficiles à se produire. Par conséquent, si un crypto-système est sécurisé contre elles, il est bien protégé contre tous les autres niveaux d'attaques (Knudsen 1994; Schneier 1994).

3. Classification des cyber-attaques selon la méthode utilisée

D'un autre point de vue, nous trouvons plusieurs familles d'attaques cryptanalytiques visant la confidentialité des données, les plus connues étant (1) la recherche exhaustive de la clé, (2) l'analyse fréquentielle, (3) la cryptanalyse linéaire et (4) la cryptanalyse différentielle.

La recherche exhaustive de la clé, nommée aussi attaque par force brute (brut force attack), teste chacune des clés possibles pour déterminer la clé qui permet la traduction de tous les messages cryptés. La complexité de cette technique réside dans le nombre de clés possibles.

L'analyse fréquentielle a été développée initialement par le mathématicien Abu Youcef Yaakob Ibn Ishaq al Kindi et reprise par E.C. Shannon, elle se base sur l'examen des fréquences d'apparition des lettres et des mots de messages cryptés afin de déterminer les clés de cryptage. Cette technique est trop efficace contre les chiffrements mono-alphabétiques présentant un biais statistique.

La cryptanalyse linéaire est inventée par le japonais Mitsuru Matsuru, elle se base sur une approximation linéaire entre les mots de sortie et les mots d'entrée pour déterminer des bits de clés.

Finalement, la cryptanalyse différentielle est initiée par Bihan et Shamir, elle consiste à comparer les sorties de deux entrées qui ne se diffèrent que d'une entrée fixe.

V. Difficulté calculatoire et confidentialité inconditionnelle.

La confidentialité d'un crypto-système dépend généralement de la complexité des algorithmes de cryptage et de la capacité calculatoire des ressources disponibles chez le cryptanalyste. Un crypto-système est dit inconditionnellement confidentiel s'il est sécurisé contre les cyber-attaques quelle

que soit la capacité calculatoire du cryptanalyste. Par analogie, Un système arithmétiquement confidentiel est tout système résistant aux cyber-attaques des cryptanalystes limités en ressources calculatoires.

Afin de mesurer la qualité d'un crypto-système, Shannon a défini la distance d'unicité n_{ud} comme le plus petit nombre de messages cryptés *C1, C2, C3…*Cn_{ud} dont un cryptanalyste peut utiliser pour définir les clés utilisées (Knudsen 1994; Shannon 1949). Autrement dit :

$$H(K/C1, C2, \dots, Cn_{ud}) = 0 \qquad (3.9)$$

Cette métrique relie l'entropie de la source des clés *H(K)* avec la redondance de la source des messages clairs $R(S_M)$ par la formule de Shannon :

$$n_{ud} = \frac{H(K)}{R(S_M)} \qquad (3.10)$$

De ce fait, plus la distance d'unicité est grande, plus le crypto-système est confidentiel.

Un crypto-système est dit inconditionnellement confidentiel lorsque sa distance d'unicité tend vers une valeur infinie.

$$\text{Le crypto} - \text{système est inconditionnellement confidentiel} \Leftrightarrow n_{ud} \to \infty \qquad (3.11)$$

En se basant sur la relation (3.10) et la définition (3.11), nous remarquons que la distance d'unicité tend vers une valeur infinie pour une valeur d'entropie de la source des clés $H(K)$ qui tend vers une valeur infinie ou pour une valeur de la redondance de la source des messages clairs $R(S_M)$ nulle. Autrement dit :

$$n_{ud} \to \infty \Leftrightarrow \begin{cases} H(K) \to \infty\ (confidentialité\ parfaite) \\ \vee \\ R(S_M) \to 0 (confidentialité\ idéale\ forte) \end{cases} \qquad (3.12)$$

La relation (3.12) définit deux types de systèmes de confidentialité inconditionnelle : Shannon a nommé le premier par système de confidentialité parfaite et le second par système de confidentialité idéale forte.

1. Crypto-système de confidentialité parfaite

Un crypto-système de confidentialité parfaite est un crypto-système inconditionnellement confidentiel avec une valeur infinie de l'entropie de la source de clés. Par conséquent, si un

cryptanalyste intercepte un message crypté par un tel système $C_i=E\ (M_i,K)$, il ne pourra pas avoir une information supplémentaire sur le message clair envoyé M_i. Nous pouvons, alors, dire que la probabilité a priori du message clair $P(M_i)$ ne diffère pas de sa probabilité a posteriori après réception du message crypté $P\ (M_i / C_i)$

$$P(M_i)=P\ (M_i / C_i) \tag{3.13}$$

Shannon a montré que le cryptage de Vernam, connu par le cryptage par masque jetable (The one time pad cipher), est un cryptage de confidentialité parfaite. Il utilise une nouvelle clé d'encryptage pour chaque nouveau message clair. Toutes les clés utilisées doivent être vraiment aléatoires, indépendantes, non répétitives et de longueur supérieure ou égale aux messages clairs. Ces caractéristiques rendent la distance d'unicité n_{ud} du cryptage de Vernam tendant vers une valeur infinie.

Le cryptage par masque jetable est résistant à toutes les cyber-attaques quel que soit le niveau d'accessibilité aux données des attaquants. Cependant, il nécessite une transmission sécurisée des clés avec une parfaite synchronisation de leurs usages. Il est vraiment difficile de satisfaire ces conditions. En pratique, les utilisateurs doivent changer les clés utilisées autant qu'ils le peuvent avec une fréquence d'utilisation de clés inférieure à *m* messages / clés définie par :

$$H(K/C1, C2, \dots Cm) \geq H_k min \tag{3.14}$$

Où $H_K min$ est la plus petite valeur de l'entropie de la source des clés pour considérer le système suffisamment incassable.

2. Crypto-système de confidentialité idéale forte

Un crypto-système de confidentialité idéale forte est un crypto-système de confidentialité inconditionnelle qui utilise des clés secrètes invariables où l'interception d'un nouveau message crypté $C_i=E\ (M_i,K)$, ne donne aucune information supplémentaire sur les clés utilisées. En effet, la probabilité a priori des clés utilisées ne diffère pas de sa probabilité a posteriori après la connaissance des messages cryptés :

$$\forall i \epsilon \mathbb{N}: H(K) = H(K/C1, C2, \dots Ci) \neq 0 \tag{3.15}$$

Un tel système est vérifié pour tout crypto-système appliqué sur une source de messages sans redondance. Dans ce cas, nous pouvons écrire :

$$\lim_{R(s_M)\to 0}(n_{ud}) = \infty \qquad (3.16)$$

Nous pouvons donc conclure que pour des messages clairs indépendants d'une source aléatoire de distribution uniforme, le crypto-système est inconditionnellement confidentiel indépendamment de la fonction d'encryptage et des clés utilisées.(Jerman-Blažič et al. 2001; Shannon 1949)

Les crypto-systèmes de confidentialité idéale forte résistent à toutes les cyber-attaques à texte chiffré seulement. L'utilisation d'une seule clé de cryptage peut affaiblir le cryptage contre les autres niveaux d'attaques. En effet, la connaissance d'un nombre suffisant de couples (messages clairs / messages cryptés) peut aider à la rupture de la confidentialité des algorithmes cryptographiques.

D'autre part, rares sont les sources de messages sans redondance, de messages facilement compressibles ou de distribution aléatoire uniforme. Par contre, la compression des données et l'ajout de messages aléatoires modifient les caractéristiques probabilistes de la source de messages pour augmenter la distance d'unicité tel qu'il est montré dans (Knudsen 1994; Schneier 1994). En effet, plus la distribution de la source tend à être uniforme plus la distance d'unicité est importante.

VI. Crypto-système de confidentialité arithmétique

En pratique, la sécurisation des systèmes ne nécessite pas l'utilisation d'un crypto-système avec une distance d'unicité de valeur infinie, mais d'un crypto-système avec une complexité supérieure aux capacités calculatoires des attaquants. La complexité d'attaque est, généralement, définie comme le nombre moyen d'opérations nécessaires pour briser la confidentialité d'un crypto-système. En d'autres mots, pour qu'un crypto-système soit arithmétiquement confidentiel, il doit résister aux attaques connues de la même façon qu'il résiste aux attaques par forces brutes.

Nous exposons dans ce qui suit des crypto-systèmes considérés comme arithmétiquement confidentiels les plus utilisés :

1. DES et 3-DES

En 1973, le bureau national de standardisation, NBS (*National Bureau of Standards*, connu aujourd'hui par le *NIST),* a exprimé le besoin au développement et à l'utilisation des crypto-systèmes pour des applications civiles. Il a publié, sur le numéro du 15 mai 1973 puis sur le numéro du 27 août 1974 du *Federal Register*, un appel d'offre pour le développement et la proposition d'un crypto-système pour des applications civiles. L'algorithme du système doit répondre à une série de

recommandations notamment(Schneier 1994) : (1) L'algorithme doit fournir un haut niveau de sécurité ; (2) il doit être facile à comprendre ; (3) sa sécurité doit résider uniquement dans la clé et il doit être ouvert et compatible aux différentes applications ;(4) d'autre part, l'algorithme doit être efficace et économiquement implémentable sur des équipements électroniques ; finalement (5) l'algorithme doit être exportable.

Suite à ces appels d'offre, un groupe promoteur d'IBM, composé de Kingston and Yorktown Heights, Roy Adler, Don Coppersmith, Horst Feistel, Edna Grossman, Alan Konheim, Carl Meyer, Bill Notz, Lynn Smith, Walt Tuchman, et Bryant Tuckerman, ont proposé une version de leurs crypto-systèmes Lucifer fonctionnant avec une clé de 128 bits. Après l'examen de la proposition par la *NSA (National Security Agency)*, on a exigé la modification de l'algorithme sans donner assez d'information sur la détermination de leurs paramètres choisis et en réduisant la taille de la clé de cryptage à 56 bits ce qui représente un niveau de sécurité inférieur. En effet, certaines analyses ont révélé que le NSA a introduit des trappes invisibles lors du développement de l'algorithme pour pouvoir accéder aux données cryptées sans la connaissance des clés (Diffie 1982; Hellman et al. 1976; Morris et al. 1977). En 1976, le NSA a organisé deux groupes de travail pour démontrer que l'algorithme est assez sécurisé (Branstad et al. 1977; Morris 1978; Morris et al. 1977). Finalement, le DES a été reconnu comme un standard fédéral le 23 novembre 1976 puis publié le 15 janvier 1977 pour utilisation dans les communications gouvernementales non classifiées.

L'algorithme d'encryptage DES commence par la formation du mot x_0 par la modification des positions des 64 bits du mot clair *m* en appliquant une fonction de permutation initiale *IP*. Le mot x_0, obtenu est traité par 16 tours de schémas de Feistel successives. Finalement le message crypté est obtenu par l'application de la fonction inverse de la transposition initiale IP^{-1}() au mot résultant des 16 tours. La Figure III.2 représente l'implémentation graphique de la partie Encryptage DES.

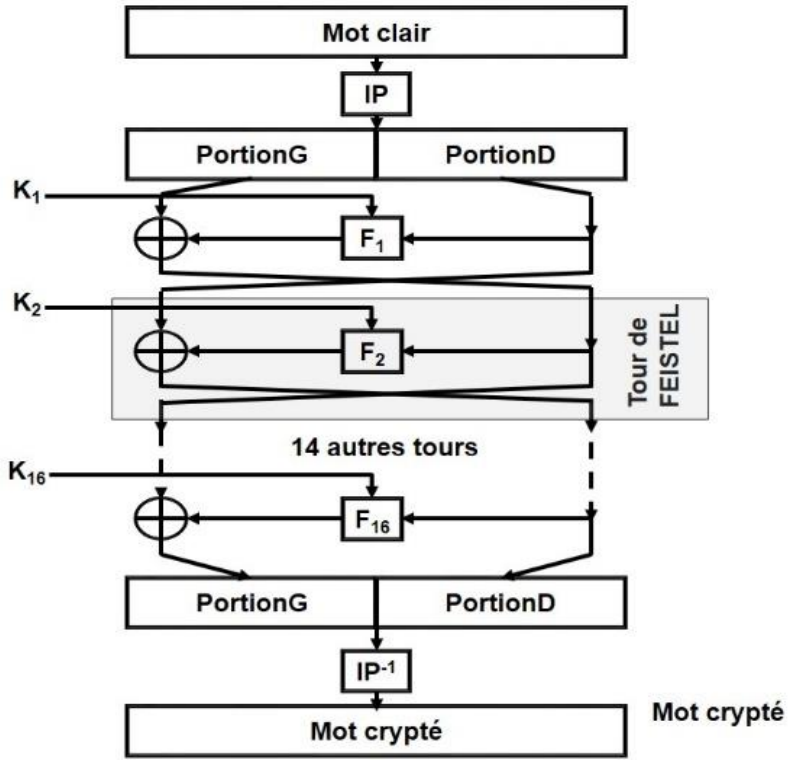


Figure III.2 Implémentation graphique de la fonction d'Encryptage DES.

Chaque $i^{ème}$ tour du schéma de Feistel commence par l'envoi de la portion droite $PortionD_{i-1}$ vers la partie gauche $PortionG_i$ de 32 bits de la sortie. La même portion droite d'entrée est traitée par une fonction de Feistel F_i avant qu'elle soit attribuée à la partie droite $PortionD_i$ de 32 bits de la sortie. Chaque fonction de Feistel F_i est paramétrée par une sous-clé k_i calculée à partir de la clé de cryptage K. F_i est définie par :

$$\begin{aligned} F_i\colon (\mathbb{Z}/2\mathbb{Z})^{32} &\longrightarrow (\mathbb{Z}/2\mathbb{Z})^{32} \\ x &\longrightarrow P\big(S(E(x)\oplus k_i)\big) \end{aligned} \tag{3.17}$$

Où E est une fonction de permutation expansive qui augmente la dimension du vecteur d'entrée de 32 bits à 48 bits en répétant certains bits et en modifiant leurs positions suivant un ordre prédéfini par le protocole.

S est une fonction de substitution composée de 8 boîtes S-box. Chaque S-box est une fonction non linéaire prédéfinie à 6 bits en entrée et 4 bits en sortie.

P est une fonction liée à la boîte de permutation P-box prédéfinie par le protocole.

k_i est la sous-clé attribuée au i[ème] tour du réseau de Feistel. Elle est calculée à partir de la clé de cryptage pour le premier tour ou à partir de la sous clé précédente par rapport aux autres tours. L'Algorithme III.1 décrit le principe de la diversification des clés pour la génération des sous-clés.

Algorithme III.1 Fonction principale de production de sous-clés Deversification()

$\boldsymbol{Input}: K\_in$ $\boldsymbol{Output}: k$ $x_0 = \boldsymbol{PC_1}(K\_in)$ $(PortionG, PortionD) = \boldsymbol{Diviser56\_28}(x_0)$ $\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ 16\ \boldsymbol{do}$ $PortionG = \boldsymbol{Left\_Shift}(PortionG, i)$ $PortionD = \boldsymbol{Left\_Shift}(PortionD, i)$ $k_i = \boldsymbol{PC_2}(\boldsymbol{Combiner28\_56}(PoritionG, PortionD), i)$ $\boldsymbol{end\ for}$ $\boldsymbol{return}\ k$	⇨ **K_in :** Clé d'entrée de 56 bits. ⇨ **k :** Vecteur de 16 clés k_i de 48 bits. ⇨ **PortionG :** les 28 bits de poids fort du mot clair ; ⇨ **PortionD :** les 28 bits de poids faible du mot clair ; ⇨ **Left_Shift :** Rotation circulaire vers la gauche d'une position pour $i \in \{1,2,9,16\}$ et de deux positions sinon. ⇨ **PC₁ :** permutation prédéfinie de $(\mathbb{Z}/2\mathbb{Z})^{56}\ vers\ (\mathbb{Z}/2\mathbb{Z})^{56}$ ⇨ **PC₂ :** permutation prédéfinie de $(\mathbb{Z}/2\mathbb{Z})^{56}\ vers\ (\mathbb{Z}/2\mathbb{Z})^{28}$ ⇨ **Diviser56_28 :** La fonction qui divise un mot 64 bits en deux mots de 32 bits. ⇨ **Combiner28_56 :** La fonction qui combine deux mots de 32 bits pour former un mot de 64 bits.

Le décryptage *DES* est assuré par l'utilisation du même schéma d'encryptage en commençant avec le mot crypté et en utilisant les clés dans le sens inverse.

L'une des critiques du DES réside dans le fait que les principes mathématiques utilisés pour déterminer les valeurs des tables de permutation et de substitution prédéfinis ont été classés top secret par l'agence américaine NSA (National security Agency). Ceci a lancé un long débat sur la sécurité du crypto-système et sur la possibilité d'introduction de failles secrètes volontaires permettant de le briser.

Une autre critique du DES est due à la longueur de sa clé considérée assez courte pour favoriser les attaques exhaustives, en particulier avec les ressources informatiques assurées par le développement de la puissance des machines actuelles et des techniques du calcul parallèle. De plus, on a prouvé que le DES présente une faiblesse contre la cryptanalyse différentielle. En juillet 1998, les trois sociétés américaines Cryptography Research, Advanced Wireless Technology et EFF ont développé la *DES Key Search Machine* qui a pu briser le *DES* et déterminer la clé de cryptage en moins de 56 heures de calcul.

Afin d'augmenter la complexité du DES, On a défini le Triple DES, noté TDES ou 3DES, comme un crypto-système construit à partir de l'association de trois blocs DES(Barker and Barker 2016; Karn et al. 1995). Il est basé sur un encryptage DES avec une clé K_1 DES suivi par un décryptage DES avec une clé 64 bits K_2 puis un encryptage DES avec une clé 64 bits K_3..Ceci peut s'exprimer par la relation :

$$TDES(m, K_1, K_2, K_3) = E(D(E(m, K_1), K_2), K_3) \quad (3.18)$$

La Figure III.3 illustre le principe de la construction du triple DES à partir de blocs DES.

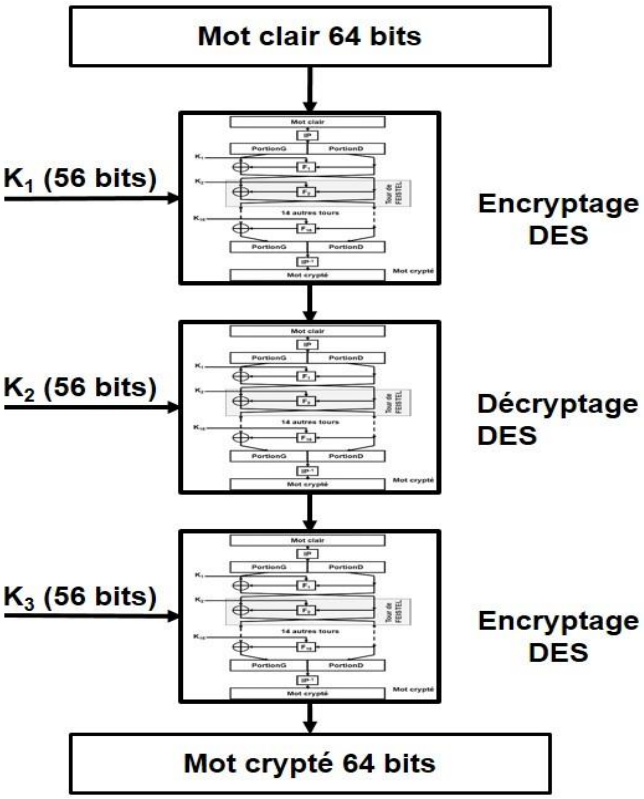


Figure III.3 Schéma bloc du crypto-système TDES.

Ce schéma permet de distinguer trois options de chiffrement en fonction de la relation entre les clés. En effet, on peut avoir l'option où les trois clés ($K_1 \neq K_2 \neq K_3$) sont toutes indépendantes, l'option où deux clés sont identiques et la troisième est indépendante des autres ($K_1 = K_3 \neq K_2$), et finalement l'option où les trois clés sont identiques. ($K_1 = K_2 = K_3$). Visuellement, la première option nécessite l'utilisation d'une clé de 168 bits (3 X 56 bits), la deuxième une clé de 112 bits et la troisième option utilise des clés de 56 bits.

L'un des inconvénients majeur de T-DES est sa vitesse qualifiée de lente, à cause de l'utilisation de 42 cellules de Feistel. Du point de vue sécurité et d'après des rapports de NIST, certaines cyber attaques ont permis la détection de fortes collisions à partir du 20ème tour du crypto-système et ils ont réduit la taille effective des clés de la première option à 112 bits et 80 bits pour la deuxième option (Barker et al. 2012).

2. Blowfish

Développé et publié en 1993 par Bruce Schneier comme une solution alternative au DES, Blowfish est un algorithme de cryptage symétrique par blocs de 64 octets avec clés de longueurs variables, de 32 à 448 bits. Son architecture modulaire le rend rapide, simple, économique et hautement sécurisé. En effet, son implémentation ne nécessite pas trop de ressources et aucune attaque connue n'a réussi à le briser à ce jour. Le Blowfish est composé de deux parties : la partie Extension de clés qui convertit les clés de cryptage de tailles inferieures à 448 bits en plusieurs sous-clés de taille globale de 4168 octets, et la partie Encryptage des données présentant la fonction d'encryptage composée de 16 tours de réseau de Feistel et utilisant un registre de clés de 18 clés de permutation de 32 bits chacune(Schneier 1993).

La Figure III.4 représente l'implémentation graphique de la partie Encryptage des données.

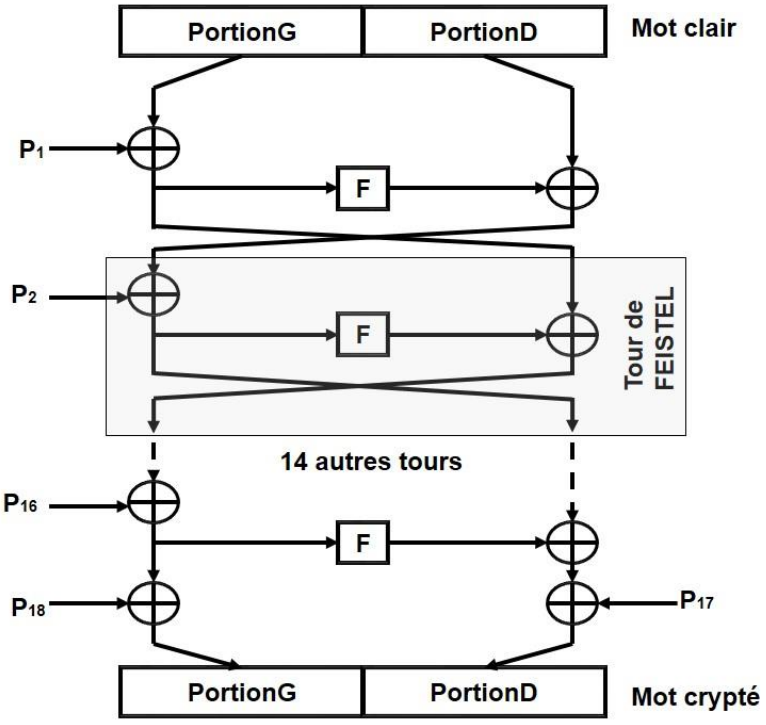


Figure III.4 Implémentation graphique de BlowFish

La fonction de Feistel *F*, définie par Blowfish, est composée de quatre boîtes S-Boxes S_1, S_2, S_3 et S_4. Elle s'applique sur les portions gauches *PortionG* de 32 bits. Elle divise la portion en quatre sous-mots de 8 bits *a, b, c* et *d.* la valeur de chaque sous-mot définie la valeur du mot de la boîte utilisée. La Figure III.5 représente l'implémentation graphique de la fonction de *Feistel* proposée par Schneier.

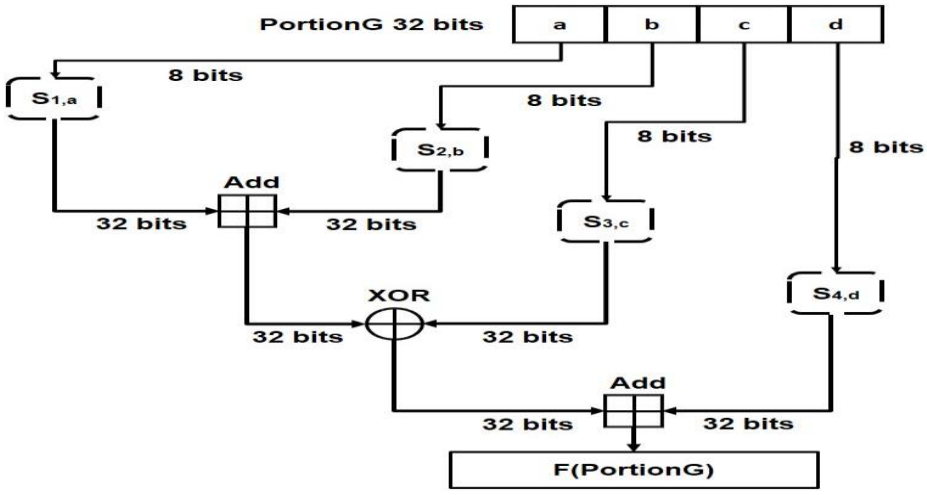


Figure III.5 Implémentation graphique de la fonction de Feistel utilisée par Blowfish.

Le décryptage du Blowfish utilise le même processus d'encryptage avec une introduction des clés de permutation dans le sens inverse.

Concernant la partie Extension de la clé, l'algorithme suivi pour produire les sous-clés est un peu lent. En effet, Il passe par sept étapes fondamentales(Schneier 1993) qui sont : (1) L'initialisation du registre des clés de permutation *P*, composé des 18 mots de 32 bits, suivi des quatre registres de substitution S_i*-boxes* avec des valeurs de la constante π exprimées en hexadécimal ;(2) La reconstruction du vecteur permutation par l'application d'une opération XOR de son contenu avec la clé de l'utilisateur. Si cette clé est courte, on doit la prolonger en la répétant ; (3) L'encryptage d'un mot nul de 64 bits avec la partie encryptage des données avec les sous-clés obtenues ; (4) Le remplacement des sous-clés P_1 et P_2 par le résultat obtenue dans l'étape (3) ; (5) L'encryptage du résultat de l'étape (3) avec les sous-clés modifiées ; (6) Le remplacement des sous-clés P_3 et P_4 par le résultat obtenue dans l'étape (5) ; (7) finalement, nous devons répéter le processus, jusqu'au remplacement de toutes les sous-clés de permutation et de substitution.

L'analyse de l'architecture du Blowfish montre que sa structure à base d'un petit nombre de blocs d'opérations simples et du réseau de Feistel le rend à la fois rapide, économique et sécurisé. Cependant, le processus de la conception des sous-clés est assez long pour résister à nombreuses attaques telles les attaques de déduction. En effet, au jour d'aujourd'hui, aucune attaque n'a réussi à affaiblir ce crypto-système.

3. Rijndael : le gagnant de l'Advanced Encryption Standard AES

Au jour d'aujourd'hui, l'évolution des ressources des ordinateurs et le développement de techniques du calcul parallèle ont causé la mort technique du DES pour les applications sensibles. En 1997, le NIST a publié un appel d'offre pour la sélection d'un successeur du DES. Le cahier des charges de l'AES exige la satisfaction des critères suivants (Standard 2001) :

- La sécurité générale ;
- Le coût en termes de calcul ;
- La simplicité, la clarté, la flexibilité, la portabilité et la facilité d'implémentation ;
- La résistance aux attaques connues ;
- Cryptage des mots de 128 bits avec des clés de 128, 192 ou 256 bits.

Après deux tours de pré-sélection (tour 0 et tour1), neuf algorithmes ont été retenus comme candidats AES: le CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97 et MAGENTA; des propriétés supplémentaires ont été posées :

- Le gagnant doit se présenter avec des versions de longueurs de clés différentes (128, 192 et 256 bits) ;
- Sa structure générale ne doit contenir que les trois opérations de base faciles à implémenter ;
- Sa performance doit être nettement supérieure à celle de DES ;
- Il doit avoir une architecture modulaire qui facilite l'implémentation parallèle ;
- Il doit fonctionner sur des équipements à faibles ressources (des processeurs de 8 bits comme des processeurs de 64 bits) ;
- Finalement, il doit résister à la cryptanalyse différentielle et linéaire (Meunier 2010).

En se basant sur ces critères, la sélection a été réduite aux cinq algorithmes finalistes : MARS (développé par une équipe d'IBM chapotée par Don Cooersmith), RC6 (Version améliorée de RC5 par Ron Rivest, Matt Robshaw, Ray Sidney, et Yiqun Lisa Yin), Rijndael (proposé par Vincent Rijmen et Joan Daemen,), Serpent (de Ross Anderson, Eli Biham, et Lars Knudsen) , et le Twofish (présenté par Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, et Niels Ferguson). En avril 2000, on a nommé le Rijndael comme vainqueur de l'AES avec un léger avancement par rapport à Twofish et Serpent(Standard 2001).

L'algorithme général du Rijndael est composé de trois phases principales : la phase initiale, les tours de transformation et la phase finale. L'algorithme copie les 128 bits du message clair dans une table d'état. Après l'ajout d'une clé du round initial, la table d'état est transformée par l'application de 10, 12 ou 14 tours en fonction de la longueur de la clé choisie. Chaque tour est composé d'une composition de fonctions élémentaires : *SubBytes*(), *ShiftRows*(), *MixColumns*() et *AddRoundKey*(). La Figure III.6 représente le schéma illustrant les différentes phases du cryptage AES.

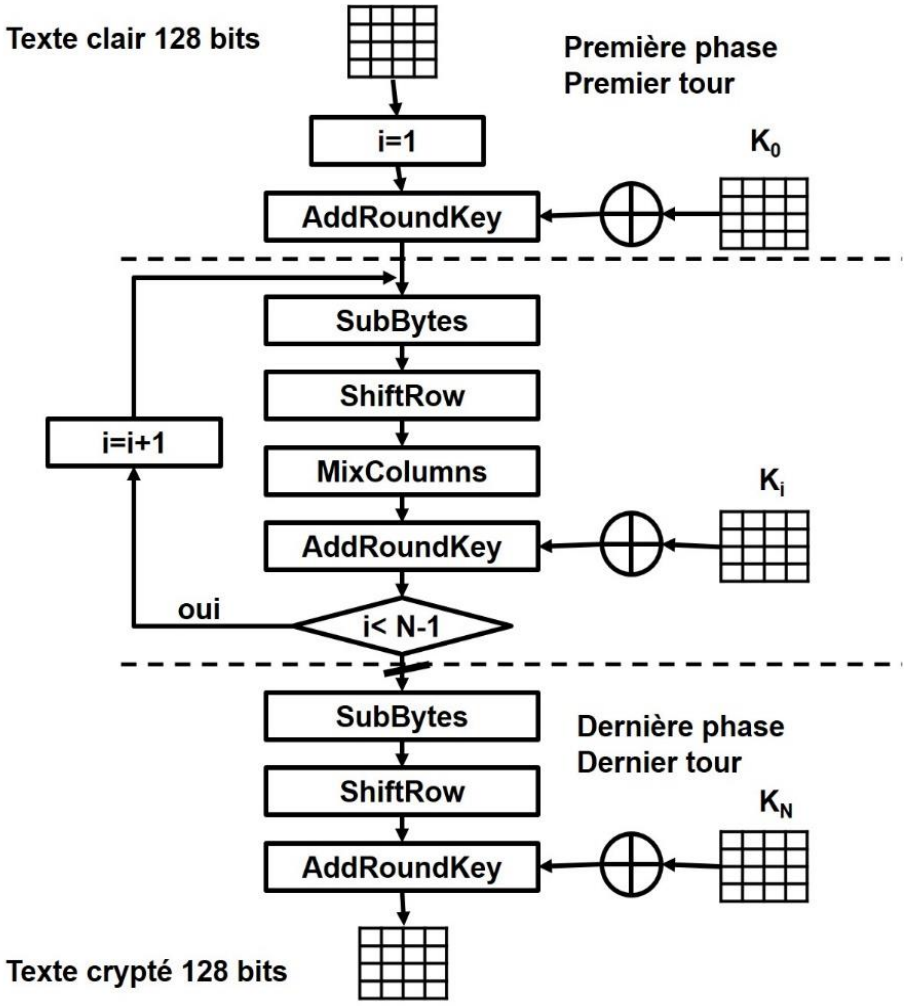


Figure III.6 Principe de cryptage AES

Au début, le message clair et les clés sont découpés en octets puis placés dans des tableaux. Le message clair contient 16 octets $a_{0,0}$, $a_{0,1}$, $a_{0,2}$...,$a_{3,3}$ classés sur une table d'état colonne par colonne. De même la clé est découpée en octets (16, 24 ou 32 octets) classés sur une table de 4 lignes. La Figure III.7 représente la construction de la table d'état et de la table de clé.

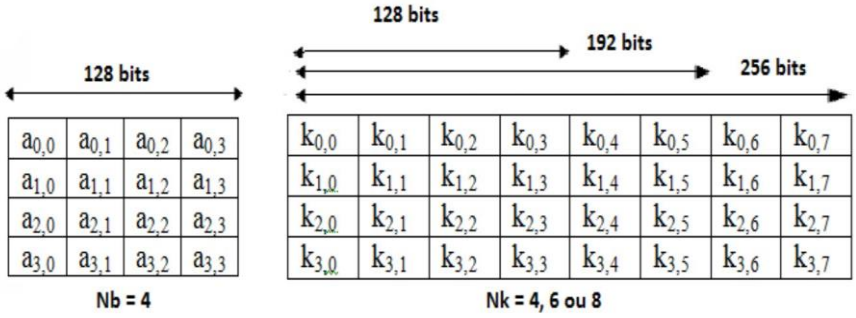


Figure III.7 Structure de la table d'état et de la table de clé.

La fonction d'addition d'une clé *AddRoundKey*() est une fonction qui additionne des sous clés aux sous blocs correspondants octet par octet en appliquant un Ou Exclusif.

La transformation *SubByte*() est une fonction de substitution non linéaire d'octet qui fonctionne indépendamment sur chaque octet de la table d'état suivant une table de substitution, dite *S-box*, prédéfinie par le standard selon la relation matricielle :

$$B' = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} B + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (3.19)$$

La Figure III.8 représente les valeurs de la table de substitution *S-box.* Si, par exemple, $S_{i,j} = 82$, la valeur de la substitution sera déterminée par l'intersection de la ligne d'indice 8 et de la colonne d'indice 2. Le résultat est :

$$S'_{i,j}=SubBytes(S_{i,j})=13 \quad (3.20)$$

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure III.8 La table de substitution S-box

La transformation décalage de ligne *ShiftRow*() est une fonction qui applique des permutations cycliques des octets sur les lignes de l'état afin d'augmenter la diffusion dans le tour, Selon la taille

des blocs message N_b. et la position de la ligne le décalage est déterminée. La Figure III.9 résume le principe de la fonction ShiftRow().

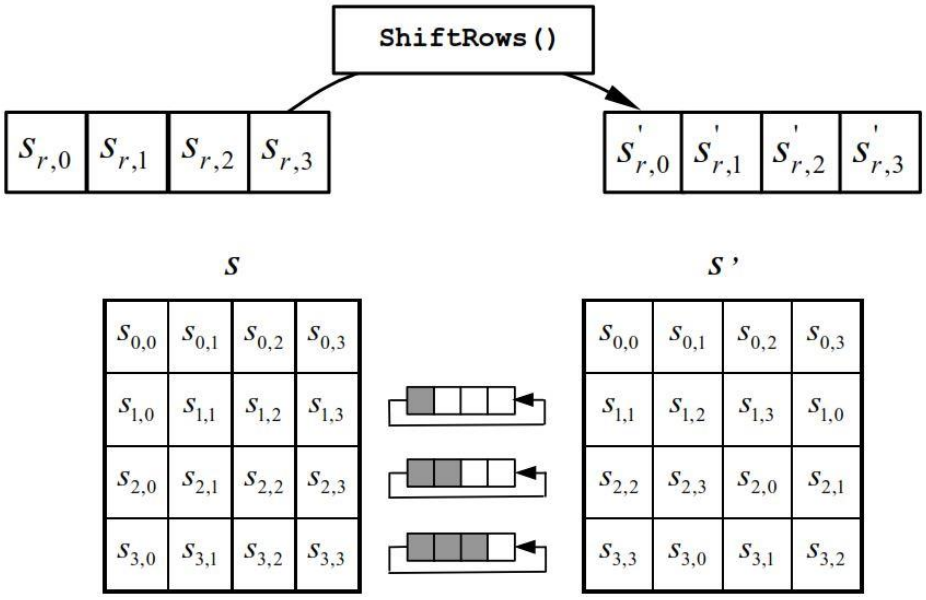


Figure III.9 Principe de la fonction ShiftRow()

La fonction de déplacement de colonne *MixColumn*() augmente d'avantage la diffusion dans le tour. Elle retourne, en sortie, les vecteurs d_i produits de la multiplication des vecteurs colonnes c_i de la table d'état par la matrice Rijndael A : $d_i = A \oplus c_i$

D'où :

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (3.21)$$

Le décryptage de l'AES se fait par l'application des fonctions *AddRoundKey*(), *InvShiftRows*(), *invSubBytes*() et *InvMixColumns*() inverse des fonctions *AddRoundKey*(), *ShiftRows*(), *SubBytes*() et *MixColumns*(). La Figure III.10 représente le principe du décryptage de l'AES.

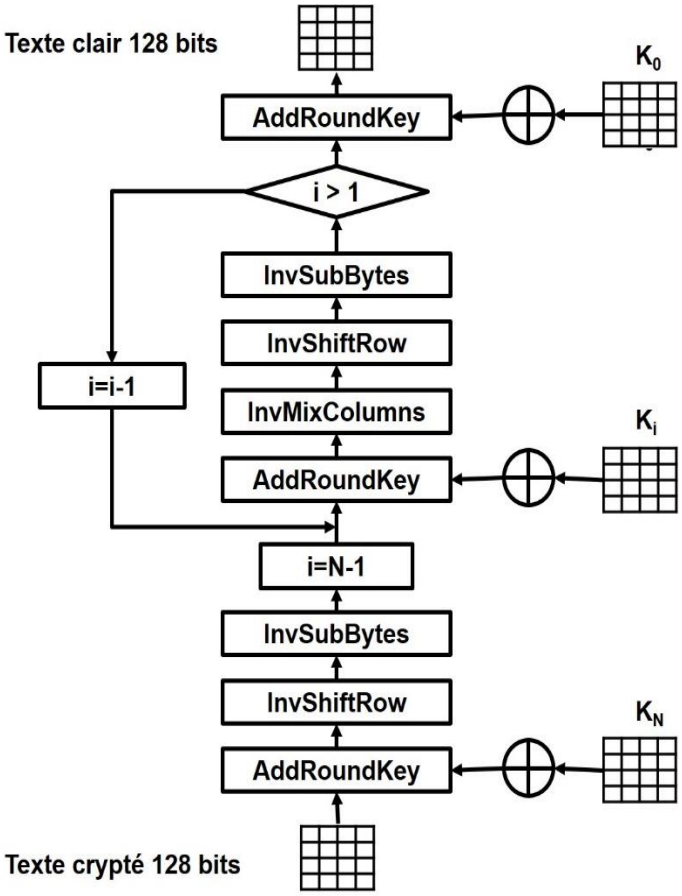


Figure III.10 Principe de décryptage de l'AES

La fonction inverse à AddRoundKey() est la fonction AddRoundKey() elle-même. La fonction inverse de la fonction ShiftRow() est la fonction InvShiftRow() qui assure des permutations cycliques vers la gauche d'un octet par rapport à la deuxième Ligne, de deux octets par rapport à la troisième ligne et finalement de trois octets par rapport à la dernière ligne.

La fonction inverse de la fonction MixColumns() est la fonction linéaire définie par la matrice inverse de A :

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} \qquad (3.22)$$

La fonction inverse de la fonction SubByte est la fonction de substitution définie par la matrice S-box inverse de la Figure III.11

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure III.11 La table inverse S-box

4. Twofish

Twofish est un algorithme de cryptage par blocs, symétrique élu comme l'un des cinq algorithmes finalistes de l'AES. Il a été développé et analysé par une équipe de cryptanalyste composée par Niels Ferguson, John Kelsey, Doug Whiting, David Wagner et Chris Hall et chapoté par le développeur du Blowfish Bruce Schneier. Bien qu'on ne l'a pas choisi comme gagnant de l'AES, certaines études asiatiques récentes le qualifie comme le finaliste le plus sécurisé et ayant le meilleur rapport performance/sécurité parmi tous les autres finalistes AES y compris le Rijndeal(Jang 2017).

D'après le document officiel présenté lors de la candidature à l'AES, Twofish crypte des mot de 128 bits avec des clés de 128 bits, 192 bits ou 256 bits avec un design simple , modulable et flexible qui permet l'utilisation d'autres longueurs de clés supérieures à 256 bits. L'algorithme fonctionne assez bien sur des processeurs 8, 16, 32 et 64 bits. Il respecte une contrainte temporelle stricte inférieure à 500 cycles d'horloge pour des machines Intel Pentium et Pentium II et il peut encrypter le mot en un temps inférieur à 10 millisecondes pour une implémentation sur processeur 8 bits avec une RAM de 64 octets (Schneier et al. 1998).

La structure du Twofish est basée sur le modèle de conception de Blowfish. Il est construit de 16 tours du schéma de Feistel qui utilise des sous-clés calculées avec une procédure complexe (*Key Schedule*), des matrices de substitutions S-boxes dépendantes des clés, de pseudo-transformation de Hadamard (PHT) et de matrices MDS (Maximum Distance Separable) assurant une diffusion binaire

maximale et efficace. La Figure III.12 représente le principe de fonctionnement de l'algorithme Twofish(Jang 2017).

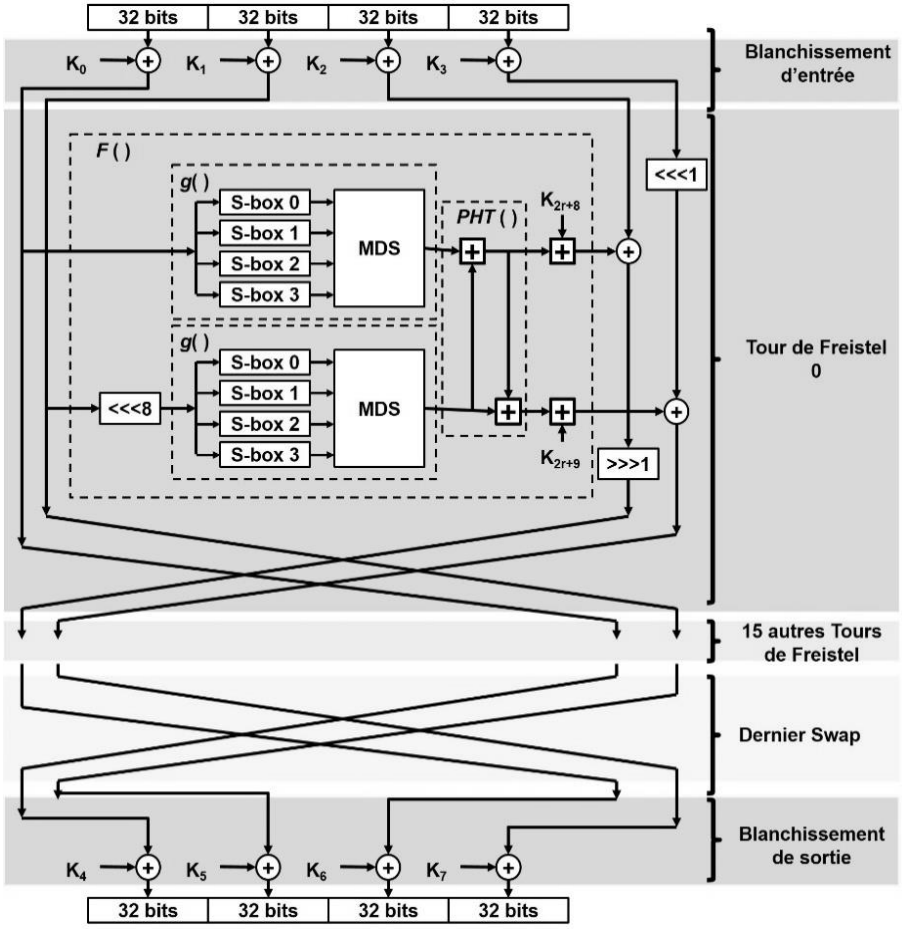


Figure III.12 Principe de fonctionnement du Twofish

L'algorithme est composé de quatre principales phases : la phase blanchissement d'entrée, 16 tours de Feistel, une SWAP, et une phase de blanchissement de sortie. L'Algorithme III.2 représente l'implémentation séquentielle du Twofish.

Algorithme III.2 Fonction principale de la partie Encryptage des Données Twofish_encrypt()

$\boldsymbol{Input}: TexteClair$	⇨ **TexteClair :** Texte Clair de 128 bits.
$\boldsymbol{Output}: TexteCrypté$	⇨ **TexteCrypté :** Texte Crypté de 128 bits.
**********$\boldsymbol{Phase\ blanchissement\ d'entrée}$**********	⇨ **Portion(*i*) :** les 32 bits du mot clair, i=0 les bits ayant le poids le plus fort, i=3 les bits ayant le poids le plus faible ;
$\big(Portion(0), Portion(1), Portion(2), Portion(3)\big) = \boldsymbol{Diviser128\_32}(TextClair)$	⇨ **Partie(*i*) :** les 8 bits de portion, i=0 les bits ayant le poids le plus haut, i=3 les bits ayant le poids le plus faible ;
$\boldsymbol{for}\ i = 0\ \boldsymbol{to}\ 3\ \boldsymbol{do}$	⇨ **Diviser128_32 :** La fonction qui
$Portion\ (i) = Portion(i)\boldsymbol{XOR}\ K(i)$	

$\boldsymbol{end\ for}$

****

**********$\boldsymbol{Phase\ 16\ Tours}$**********

$\boldsymbol{for}\ i = 0\ \boldsymbol{to}\ 15\ \boldsymbol{do}$

$Portion\ (1) = decalG(Portion(1), 8)$

$\boldsymbol{for}\ j = 0\ \boldsymbol{to}\ 1\ \boldsymbol{do}$

**********Fonctio*n(portion(0)) et fonction g(portion(1))*******

$\big(Partie(0), Partie(1), Partie(2), Partie(3)\big) = \boldsymbol{Diviser32\_8}\big(Portion(j)\big)$

$\boldsymbol{for}\ k = 0\ \boldsymbol{to}\ 3\ \boldsymbol{do}$

$Partie(k) = \boldsymbol{Sbox}(Partie(k), k)$

$\boldsymbol{end\ for}$

**********Fonction *MDS()*$**********$

$$\begin{pmatrix} Partie(0) \\ Partie(1) \\ Partie(2) \\ Partie(3) \end{pmatrix} = \begin{pmatrix} \boldsymbol{01} & \boldsymbol{EF} & \boldsymbol{5B} & \boldsymbol{5B} \\ \boldsymbol{5B} & \boldsymbol{EF} & \boldsymbol{EF} & \boldsymbol{EF} \\ \boldsymbol{EF} & \boldsymbol{5B} & \boldsymbol{01} & \boldsymbol{EF} \\ \boldsymbol{EF} & \boldsymbol{01} & \boldsymbol{EF} & \boldsymbol{5B} \end{pmatrix} \begin{pmatrix} Partie(0) \\ Partie(1) \\ Partie(2) \\ Partie(3) \end{pmatrix}$$

$G(j) = Combiner8\_32\big(Partie(0), Partie(1), Partie(2), Partie(3)\big)$

****

****

$\boldsymbol{end\ for}$

********** Fonction *PHT()*$**********$

$G(0) = \big(G(0) + G(1)\big) mod\ 32$

$G(1) = \big(G(0) + G(1)\big) mod\ 32$

****

$G(0) = \Big(G(0) + K\big((2.i) + 8\big)\Big)\ mod\ 32$

$G(1) = \Big(G(1) + K\big((2.i) + 9\big)\Big)\ mod\ 32$

$Portion\ (2) = decalG\ \Big(\big(Portion(2)\boldsymbol{XOR}\ G(0)\big), -1\Big)$

$Portion\ (3) = decalG(Portion(3), 1)\boldsymbol{XOR}\ G(1)$

**********Fonction *SWAP()*$**********$

$A = Portion(1)$

divise un mot 128 bits en quatre mots de 32 bits.

⇨ **Combiner32_128 :** La fonction qui combine quatre mots de 32 bits pour former un mot de 128 bits.

⇨ **Diviser32_8 :** La fonction qui divise un mot 32 bits en quatre mots de 8 bits.

⇨ **Combiner8_32 :** La fonction qui combine quatre mots de 8 bits pour former un mot de 32 bits.

⇨ **DecalG(m,n) :**la fonction de décalage cyclique de n bits du mot m vers la gauche, n négative implique un décalage à droite.

⇨

$Portion\ (1) = Portion(3)$

$Portion\ (3) = A$

$A = Portion(4)$

$Portion\ (4) = Portion(2)$

$Portion\ (3) = A$

****

end for

**********La phase dernier *SWAP()********************

$A = Portion(1)$

$Portion\ (1) = Portion(3)$

$Portion\ (3) = A$

$A = Portion(4)$

$Portion\ (4) = Portion(2)$

$Portion\ (3) = A$

****

***********Phase blanchissement d'entrée**********************

For $i = 0$ ***to*** 3 ***do***

$Portion\ (i) = Portion(i)$ ***XOR*** $K(i + 4)$

end for

$TexteCrypté = \boldsymbol{Combiner32\_128}(Porition0, Portion1, Portion2, Portion3)$

****

return $TexteCrypté$

L'une des forces de l'algorithme réside dans le processus de génération des sous-clés. Le processus de génération de sous-clés produit 40 sous-clés pseudo-indépendantes et quatre boîtes de substitution S-boxes à partir d'une clé de longueur *N = 128, 192* ou *256* bits. Si la clé de cryptage a une longueur inférieure à 256 bits, elle doit être complétée par des zéro jusqu'à l'atteinte d'une longueur de clé déjà définie.

L'autre point de force de ce crypto-système est l'utilisation de fonctions de substitution S-boxes dépendantes des clés de cryptage, chose qui augmente visiblement la complexité de sa cryptanalyse.

Du point de vue sécurité, rares sont les travaux qui ont pu proposer une cryptanalyse du Twofish. En 1999, Niels Ferguson a publié une attaque différentielle impossible qui brise six tours sur 16 de la version de clé de 256 bits en utilisant 2^{256} étapes (plus de 1.15 X 10^{77} étapes) (Ferguson 1999). En 2000, on a publié qu'il était possible de briser le Twofish avec une probabilité de $2^{-57.3}$ (5.67 X 10^{-18}) en utilisant 2^{51} (2.25 X 10^{15}) couples de (messages clairs/ cryptés) connus. Ceci a été démontré par Scheiner où il a montré que cette négligeable probabilité ne peut être atteinte en exploitant uniquement 2^{51}couples(Schneier 2005).

5. Threefish

Threefish est un crypto-système développé en 2008 par Bruce Schneier, Niels Ferguson et toute l'équipe de Skein (l'algorithme de hachage finaliste de SHA-3). Basé sur trois opérations fondamentales : l'addition modulo 2^{64}, la permutation et le XOR appliquées sur des mots de 64 bits(Ferguson et al. 2010), le Threefish crypte des mots clairs de taille 256, 512, ou 1024 bits avec des clés de même tailles. Il utilise 72 tours pour crypter des mots de 256 ou de 512 bits et 80 tours pour des mots de 1024 bits. Un bloc Threefish contient quatre tours Threefish qui finit par un ajout d'une sous clé de cryptage. La Figure III.13 représente l'algorithme général du Threefish 512.

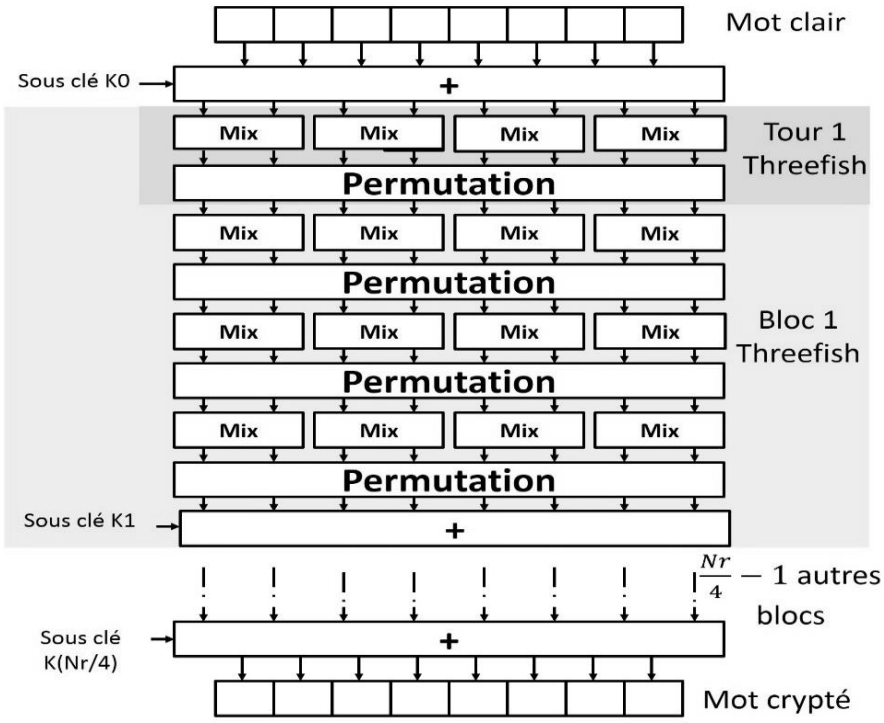


Figure III.13 Algorithme général du Threefish

Les développeurs de Threefish ont montré que la non-linéarité du système est assurée par les bits d'ajout par addition, et qu'une diffusion totale des bits du mot clair est vérifiée à partir du 9ème tour du Threefish-256, 10ème tour du Threefish-512 et le 11ème tour du Threefish-1024. Chaque tour est construit de deux niveaux : le premier niveau contient des fonctions Mix (4 fonctions pour le Threefish-256, 8 fonctions pour le Threefish-512 et 16 fonctions pour le Threefish-1024). Le deuxième niveau contient une fonction de permutation (Ferguson et al. 2010).

Les fonctions Mix sont des fonctions à deux mots de 64 bits en entrée (x_0,x_1) et de deux mots de 64 bits en sortie (y_0, y_1) telle que :

$$\begin{cases} y_0 = (x_0 + x_1) mod\ 2^{64} \\ y_1 = (x_1 \ll R_{(d\ mod\ 8),j}) \oplus y_0 \end{cases} \quad (3.23)$$

Avec << est une rotation à gauche avec une constante $R_{d,j}$ où d représente le numéro du tour et j représente la position de la fonction Mix. Les valeurs de $R_{d,j}$ sont représentées dans le Tableau III-1

Tableau III-1 Valeurs des constantes de rotation en fonction du numéro de tour et de la position de la fonction Mix

Threefish		256		512				1024							
J		0	1	0	1	2	3	0	1	2	3	4	5	6	7
d=	0	14	16	46	36	19	37	24	13	8	47	8	17	22	37
	1	52	57	33	27	14	42	38	19	10	55	49	18	23	52
	2	23	40	17	49	36	39	33	4	51	13	34	41	59	17
	3	5	37	44	9	54	56	5	20	48	41	47	28	16	25
	4	25	33	39	30	34	24	41	9	37	31	12	47	44	30
	5	46	12	13	50	10	17	16	34	56	51	4	53	42	41
	6	58	22	25	29	39	43	31	44	47	46	19	42	44	25
	7	32	32	8	35	56	22	9	48	35	52	23	31	37	20

La fonction de permutation entre les portions de 64 bits est définie de façon à maximiser la distance entre eux et assurer une bonne diffusion. Le Tableau III-2 définit la permutation des portions en fonction de la version de Threefish et de la position de la portion.

Tableau III-2 Définition de la fonction de permutation du Threefish.

		Position de la portion de 64 bits															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Threefish	256	0	3	2	1												
	512	2	1	4	7	6	5	0	3								
	1024	0	9	2	13	6	11	4	15	10	7	12	3	14	5	8	1

L'algorithme Threefish est composé de *Nr* blocs. A la fin de chaque bloc, l'algorithme additionne aux mots traités une sous-clé dérivée de la clé de cryptage *Kc* introduite (ayant la taille du mot crypté) et d'un mot Tweak de 128 bits. L'Algorithme III.3 définit le processus de génération des sous-clés.

Algorithme III.3 Algorithme de génération de sous-clés

$\boldsymbol{Input}: Kc, Tweak$

$\boldsymbol{Output}: K$

$(t_1, t_0) = \boldsymbol{Diviser128\_64}(Tweak)$

$t_2 = t_1 \oplus t_0$

$(k_{Nw-1}, k_{Nw-2}, \dots, k_0) = \boldsymbol{Diviserx - 64}(\boldsymbol{Kc})$

$k_{Nw} = C_{240} \oplus_{i=0}^{Nw-1} k_i$

$\boldsymbol{For}\ s = 0\ \boldsymbol{to}\ \frac{Nr}{4} - 1\ \boldsymbol{do}$

$\boldsymbol{For}\ i = 0\ \boldsymbol{to}\ Nw - 4\ \boldsymbol{do}$

$k_{s,i} = k_{(s+i) mod (Nw+1)}$

$\boldsymbol{end\ for}$

$k_{s,Nw-3} = k_{(s+Nw-3) mod (Nw+1)} + t_{s \bmod 3}$

$k_{s,Nw-2} = k_{(s+Nw-2) mod (Nw+1)} + t_{(s+1) \bmod 3}$

$k_{s,Nw-1} = k_{(s+Nw-1) mod (Nw+1)} + s$

$K_s = \boldsymbol{Combiner64\_x}(k_{s,Nw-1}, k_{s,Nw-2}, \dots, k_{s,0})$

$\boldsymbol{end\ for}$

$\boldsymbol{return}\ K$

⇨ **x :** version de Threefish 256, 512 ou 1024 ;
⇨ **Kc :** Clé de *x* bits ;
⇨ **C_{240}**=0X1BD11BDAA9FC1A22 choisi pour s'assurer que les sous-clés ne soient pas nulles
⇨ **Tweak :** mot de 128 bits ;
⇨ **Nw :** La taille de la clé / 64 ;
⇨ **Nr :** nombre de tours ;
⇨ **K :** Le vecteur de Nr/4 sous-clés ;
⇨ **Diviser A_B :** La fonction qui divise un mot A bits en quatre mots de B bits ;
⇨ ***Combiner64_x*** = La fonction qui combine des mots de 64 bits pour former un mot de x bits.

Le décryptage de Threefish se fait par le chemin inverse de l'encryptage en introduisant les sous-clés par un ordre inverse à celui de l'encodage et en utilisant des fonctions de permutation et de mixage inverses.

Il est à noter qu'à ce jour, aucune attaque publiée n'a pu briser plus de 35 tours de Threefish de la version 1.3 (Aumasson et al. 2009; Ferguson et al. 2010) développé en se basant sur les critères de (Khovratovich and Nikolić 2010).

L'analyse de Threefish montre qu'à base d'opérations simples (addition, XOR et permutation) et sans l'utilisation de boîtes de substitution ou de schéma de Feistel, on a développé un crypto-système hautement sécurisé avec une totale diffusion des bits et une non-linéarité due aux bits portés par l'addition.

VII. Développement d'un Protocole SCADA Sécurisé

Les systèmes SCADA présentent des qualités fonctionnelles importantes suite à leurs architectures centralisées adaptées et aux protocoles de transport de données simples et riches en services d'application. Du point de vue sécuritaire, nous avons vu que ces protocoles sont des protocoles transparents sans aucun cryptage intégré. La sécurisation de ces protocoles par des crypto-systèmes conçus typiquement pour des systèmes IT n'est pas possible vue les différences entre les spécificités des deux systèmes. Afin de remédier à cette problématique, notre analyse des caractéristiques des systèmes SCADA et des crypto-systèmes décrits précédemment nous a montré que nous devions développer un nouveau crypto-système qui offre à la fois :

- Un respect de tous les principes de Kerckhoff ;
- Un très haut niveau de sécurité qui tend vers une sécurité inconditionnelle et assure la confidentialité, l'intégrité et l'authentification des trames inter changées ;
- Un fonctionnement en temps réel du système SCADA sur le même support de communication ;
- Une préservation de l'architecture et des performances fonctionnelles des protocoles sécurisés ;
- Une possibilité d'implémentation sur micro-processeur ou sur FPGA ;
- Une architecture modulaire qui favorise l'implémentation parallèle du crypto-système ;
- Une structure simple basée sur des opérations logiques de base (Permutation et XOR) ;

- La complexité du système et sa force doit être dans le processus de la génération des sous-clés utilisées.

En se basant sur ces recommandations, nous avons choisi de concevoir un nouveau crypto-système qui implémente les systèmes théoriques de confidentialité idéale forte et de confidentialité parfaite de Shannon. Dans ce cadre, nous avons proposé la modification des caractéristiques probabilistes des sources d'information en ajoutant un bruit aléatoire de caractéristiques étudiées pour assurer la confidentialité idéale forte et l'utilisation de sous-clés pseudo-aléatoires et pseudo-indépendantes dans le cryptage d'un nombre très limité de trames pour implémenter la confidentialité parfaite. Ces sous-clés doivent être calculées au préalable et organisées dans des tables simples de petites tailles (de l'ordre de Kilo-octets). La réponse temporelle du crypto-système peut être réduite en le construisant à partir d'un nombre réduit de tours (Permutation et XOR) traitant les trames en un seul bloc.

Comme pour les crypto-systèmes développés par Bruce Schneier, la force de notre crypto-système doit résider dans la procédure du traitement des clés et de la génération des sous-clés. De ce fait , (1) le crypto-système doit accepter l'utilisation de clés de tailles variables ; (2) la procédure de génération des sous-clés doit être complexe, lente et indépendante des crypto-systèmes intermédiaires ; (3) les sous-clés utilisées doivent être pseudo-aléatoires, pseudo-indépendantes et d'une taille similaire à la taille de la trame à crypter ;finalement (5) La procédure de génération de sous clés doit être basée sur des algorithmes de hashage et de générateur de nombres pseudo-aléatoires de haute sécurité.

VIII. Conclusion

Afin de définir les caractéristiques du crypto-système employé dans le développement de notre protocole de transport SCADA sécurisé, nous nous sommes intéressés à l'étude des principes fondamentaux de la cryptographie moderne, à la classification des cyber-attaques et à l'analyse fonctionnelle des crypto-systèmes classés, à ce jour, comme systèmes sûrs. Le crypto-système développé doit être clair, adapté aux spécificités des systèmes SCADA, applicable sur les protocoles de transport de données, hautement sécurisé et facilement implémentable, basé sur des opérations simples et rapides.

Nous allons présenter, dans le prochain chapitre, l'architecture détaillée du ST-101, notre nouveau protocole SCADA hautement sécurisé. Du point de vue fonctionnel, nous nous sommes basés, dans sa conception, sur la structure de l'IEC 60870-5-101, le protocole SCADA non routable le plus utilisé dans la gestion des réseaux électriques en Algérie et en Europe, qualifié à la fois de simple et riche en fonctions.

CHAPITRE IV : LE ST-101 : Le protocole de transmission SCADA sécurisé

I. Introduction

Le protocole SCADA IEC 60870-5-101 est le protocole de transmission de données le plus utilisé en Algérie pour la téléconduite des réseaux électriques. Afin de lui assurer un haut niveau de sécurité, nous avons introduit une couche de Sécurité entre la couche physique et la couche liaison de son architecture avancée. Cette couche implémente le principe de la confidentialité inconditionnelle de SHANNON. Elle chiffre, principalement, les trames IEC-101 de longueurs variables afin de cacher toutes les informations communiquées en s'assurant de l'authenticité des équipements et de l'intégrité des données. Notre approche doit protéger le système, contre les attaques passives et les attaques par modification et fabrication des messages, sans modifier les caractéristiques techniques de sa couche physique ou de la longueur optimale de ses trames et ASDU.

Afin d'atteindre notre objectif, nous avons conçu la couche de sécurité à partir de deux sous-couches : La sous-couche Egalisateur de la distribution et la sous-couche Conception du cryptogramme. La Figure IV.1 représente le diagramme en blocs de la couche Sécurité.

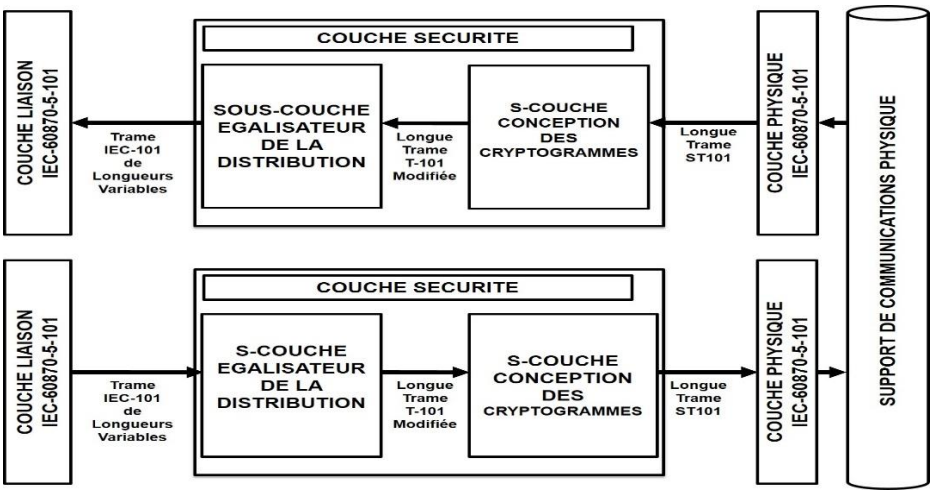


Figure IV.1 Diagramme en blocs de la sous couche Sécurité

II. Sous-couche égalisateur de la distribution

En se basant sur l'implémentation pratique de la confidentialité idéale forte de SHANNON (Knudsen 1994; Shannon 1949), la sous-couche Egalisateur de la distribution modifie les caractéristiques probabilistes des trames IEC-60870-5-101 standards et influe sur sa distribution pour la faire converger vers une distribution uniforme afin de satisfaire les propriétés de la confidentialité idéale forte. Elle transforme les trames IEC-60870-5-101 de longueurs variables en de longues trames T-101 modifiées : Elle supprime la redondance dans les en-têtes des trames standards et ajoute des octets de bruit de valeurs aléatoires afin d'obtenir des trames de 261 octets de longueur.

1. Notion de base

Les systèmes SCADA gèrent des systèmes temps réel de missions critiques, ils doivent par conséquent respecter les contraintes temporelles importantes. La couche physique de l'IEC 60870-5-101 est conçue spécialement pour assurer des transmissions fiables et continues des trames de longueur maximale de 261 octets. En réalité, ces trames ne peuvent atteindre que rarement cette longueur maximale.

Afin d'étudier les longueurs des trames IEC-60870-5-101 échangées entre les MTU et les RTU, nous avons analysé les trames IEC-101 du système SCADA utilisé par la Société de Distribution de l'Electricité et du Gaz d'Alger (SDA) pour la gestion de son réseau électrique de distribution. Elles sont capturées entre le mois d'Avril 2013 et le mois de Janvier 2014 au niveau des deux ports série RS-232 de la MTU : Le port 5 et le port 36. Le Tableau IV-1 représente les caractéristiques techniques de ces deux ports.

Tableau IV-1 Caractéristiques techniques des ports 5 et 36 du système SCADA de la SDA

Caractéristiques	***Port 5***	***Port 36***
Le Protocole SCADA	IEC-60870-5-101	IEC-60870-5-101
Débit du port RS-232	9600 bps	9600 bps
Trames RS-232	8 bits data/ parité paire/ 1 bit stop	8 bits data/ parité paire/ 1 bit stop
Contrôle du flux	RTS/ CTS	RTS/ CTS
Support de communication	Fibre optique	Système WLL SR-500
Configuration du réseau	Point à point	Point à multipoints
Mode de transmission	Asymétrique	Asymétrique
Mode d'interrogation	Continue	Continue
Nombre de postes électriques contrôlés	01	05
Nombre de RTU connectées	01	06
Nombre d'adresses d'objets	2385	12019

Cette analyse a révélé que la longueur moyenne des trames IEC-101 de longueurs variables collectées du port 5 était de 18.96 octets avec un écart type de 29.37 octets ; alors que la longueur moyenne des trames de longueurs variables collectées du port 36 était de 24.96 octets avec un écart type de 24.47 octets. D'autre part, la longueur moyenne journalière la plus longue était de 63.32 octets enregistrée au niveau du port 36. La Figure IV.2 représente la distribution des longueurs des trames de longueurs variables capturées des ports 5 et 36.

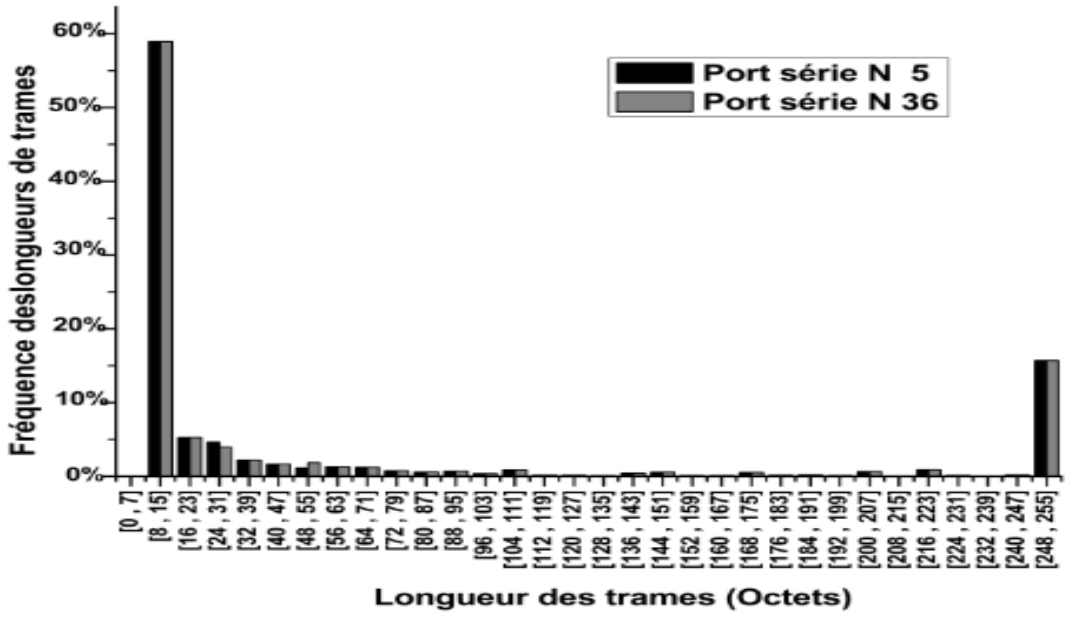


Figure IV.2 Distribution des longueurs des trames de longueurs variables

Cette distribution nous montre que plus de 75% des trames de longueurs variables sont des trames courtes de longueurs inférieures à 50 octets par trame. Ce résultat illustre la possibilité d'ajouter des octets de bruit aux trames afin de modifier la distribution de la source des messages sans affecter le fonctionnement en temps réel du système SCADA. Ceci nous a motivé à concevoir la sous couche Egalisateur de la distribution à partir de deux blocs : Le bloc Additionneur du bruit adaptatif et le bloc Modificateur de la structure des trames.

La Figure IV.3 représente la structure de l'Egalisateur de la distribution. Lorsque la couche Liaison envoie une trame IEC-60870-5-101 de longueur variable inférieure à 261 octets, l'Additionneur du bruit adaptatif transforme cette trame en une trame longue de 261 octets par l'ajout de (254-*L*) octets de bruit et d'un octet supplémentaire dit octet de réserve, où (*L*) représente la longueur du corps de la trame IEC-101. Le Modificateur de la structure de la trame modifie l'en-tête standard de la trame IEC-101 de longueur variable en supprimant la redondance dans les troisièmes et les quatrièmes

octets des trames et en les remplaçants par deux octets représentant les clés de synchronisation utilisées par la sous-couche Conception des cryptogrammes.

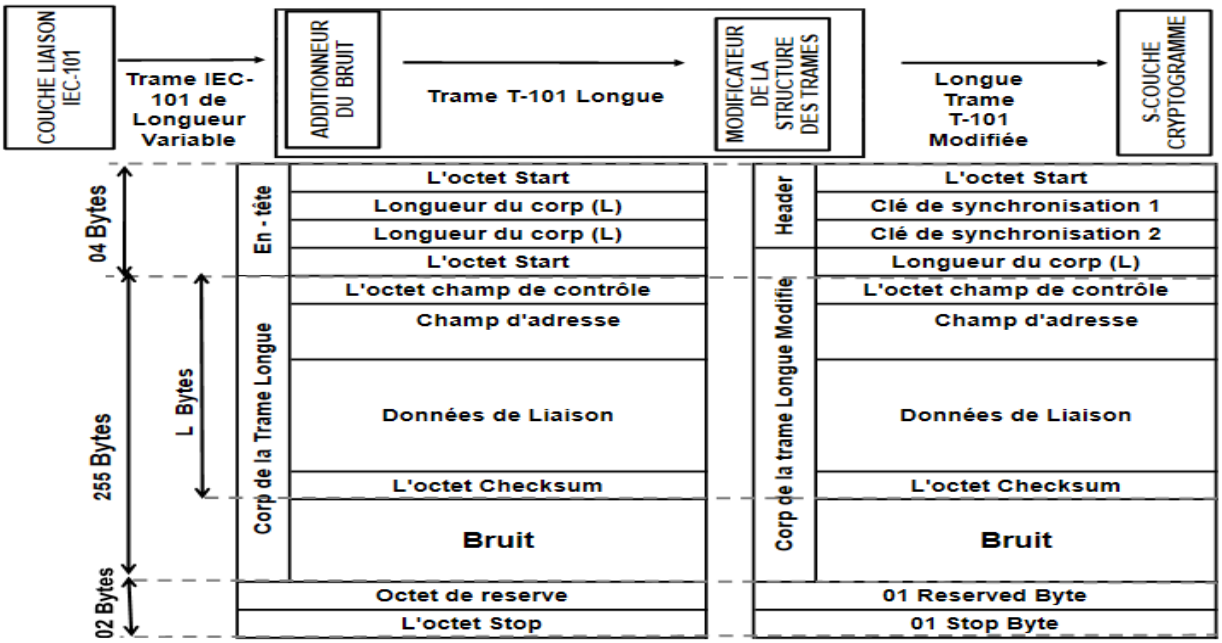


Figure IV.3 Structure de l'Egalisateur de la distribution

2. Bloc Additionneur du bruit adaptatif

L'analyse profonde des caractéristiques probabilistes des distributions est une approche très commune dans la cryptanalyse. De fait, l'analyse des trames IEC-101 capturées a révélé que certaines valeurs des octets sont plus fréquentes que d'autres. Afin de fabriquer une source de trames ayant une distribution uniforme, le bruit ajouté doit avoir une distribution qui dépend, à la fois, de la longueur moyenne des trames, de l'espace d'échantillonnage et de la distribution des trames standards.

Soit *N* la longueur moyenne des trames IEC-101 de longueur variable et soit *B* une variable aléatoire d'une fonction de distribution $P(b)$, où b est dans l'espace d'échantillonnage $\Omega_m = \{b_0, b_1, b_2 \dots b_{n-1}\}$ correspondant à l'ensemble des *n* valeurs d'octets contenus dans les trames. Pour obtenir une distribution uniforme de la source de trames envoyées S_m, chaque octet b_i doit avoir une fréquence d'apparition égale aux fréquences des autres octets. Ceci nécessite l'ajout d'un bruit formé de N' octets avec une fonction de distribution $P'(b_i)$.

Soit $P_i = P(b_i)$ et $P_i' = P'(b_i)$.

S_m a une distribution uniforme $\Rightarrow \forall i \in \{0,1,2,\dots,(n-1)\}: N.P_i + N'P'_i = \frac{N+N'}{n}$ (4.1)

Autrement dit :

S_m a une distribution uniforme

$$\Leftrightarrow \forall i \in \{0,1,2,\dots,(n-1)\}: P'_i = \frac{1}{N'}\left(\frac{N+N'}{n} - (N.P_i)\right) \quad (4.2)$$

Sachant que $P'_i \in [0,1]$. On définit N'_{min}comme la plus petite longueur de bruit additionné qui permet l'obtention d'une source de message S_m uniformément distribuée. On a :

$$\forall i \in \{0,1,2,\dots,(n-1)\}: 0 \leq P'_i \leq 1 \quad (4.3)$$

Ou encore :

$$\forall i \in \{0,1,2,\dots,(n-1)\}: 0 \leq \frac{1}{N'}\left(\frac{N+N'}{n} - (N.P_i)\right) \leq 1 \quad (4.4)$$

Par conséquent on aura :

$$N'_{min} = max\left(max_i\left(n.N\left(P_i - \frac{1}{n}\right)\right), N\right) \quad (4.5)$$

Autrement dit, quelle que soit la distribution initiale de la source des messages, l'ajout d'un bruit d'octets aléatoires augmente la distance d'unicité du cryptage utilisé indépendamment de la distribution du bruit(Knudsen 1994). Par contre, On peut implémenter le principe de la confidentialité idéale forte et faire converger la distance d'unicité vers une valeur infinie en ajoutant, au minimum, un bruit de longueur moyenne de N'_{min} octets avec une fonction de distribution P' définie par les relations (4.2) et (4.6).

En se basant sur les deux relations précédentes, nous avons développé une implémentation modulaire de la sous-couche Egalisateur de la distribution. La Figure IV.4 illustre l'architecture que nous avons adopté pour cette implémentation.

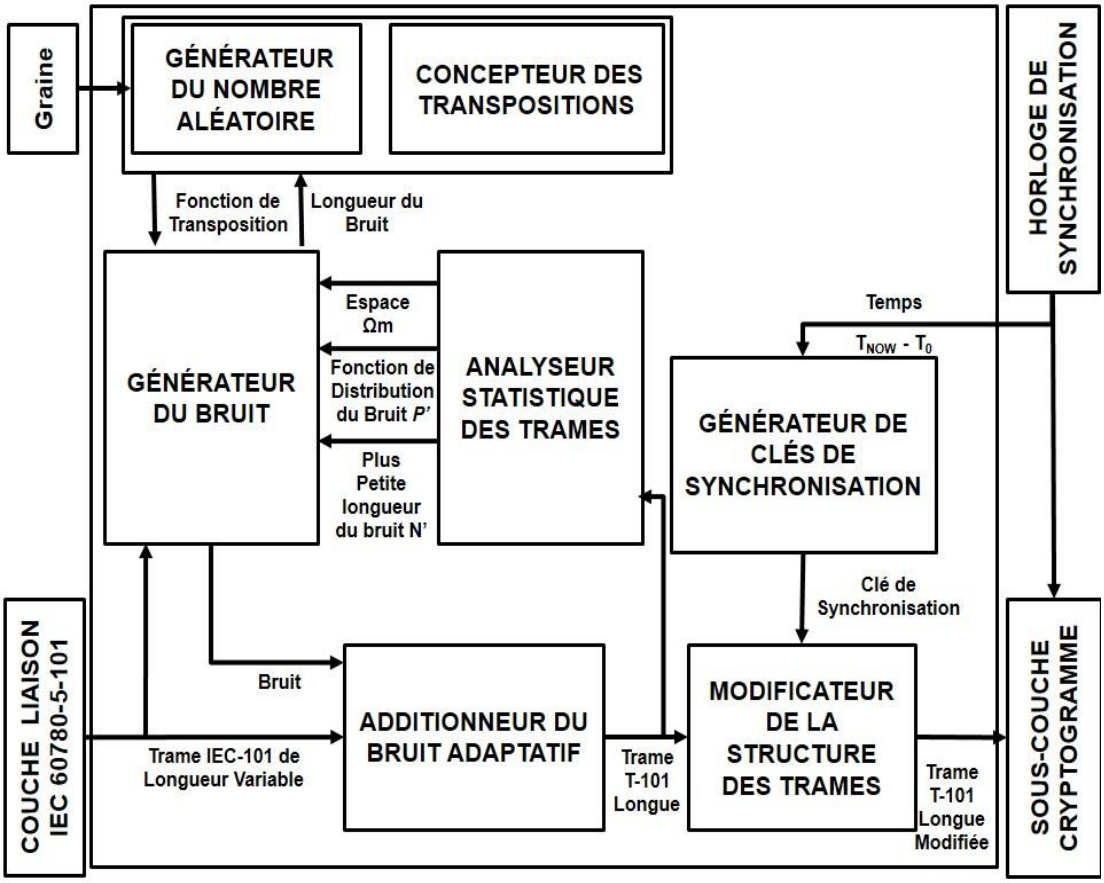


Figure IV.4 Architecture de la sous couche Egalisateur de la distribution.

Le bloc Analyseur statistique des trames détermine les paramètres statistiques du bruit additionné :**(1)** Il analyse les corps des précédentes trames IEC-101 longues ;**(2)** il définit l'espace d'échantillonnage des trames Ω_m et la fonction de distribution $P(b)$; **(3)** il calcule la longueur moyenne des trames IEC-101 de longueur variable *N* (en se basant sur les valeurs des deuxièmes et troisièmes octets des en-têtes) ; **(4)** et finalement, il calcule les paramètres du bruit ajouté, à savoir: l'espace d'échantillonnage du bruit $\Omega_N = \Omega_m$, la fonction de distribution du bruit $P'(b)$et la plus petite longueur du bruit ajouté N'_{min}.

En se basant sur les paramètres du bruit ajouté, le bloc Générateur du bruit produit une nouvelle séquence pseudo-aléatoire du bruit pour chaque nouvelle trame IEC-101 de longueur variable. Tout d'abord, il produit une séquence préliminaire du bruit par la répétition des éléments de l'espace d'échantillonnage du bruit $\Omega_N$respectivement à la fonction de distribution du bruit *P'(b)*. Puis, il réarrange les octets de la séquence à base d'une transposition (permutation) aléatoire T_i définie par le

bloc Concepteur des transpositions. Finalement, il transmet les (*254-L*) octets du bruit au bloc Additionneur du bruit pour les ajouter à la trame IEC-101 de longueur variable.

L'Algorithme IV.1 représente le pseudo code de la fonction principale du bloc Générateur du bruit adaptatif.

Algorithme IV.1 Algorithme du bloc Générateur du bruit adaptatif

Input : $EspaceEchantillon(i), F(i) \quad (i = 1,2 \dots N)$
Output : $VecteurBruit$
$l = 0$
for $i = 0$ **to** N **do**
for $j = 1$ **to** $F(i)$ **do**
$l = l + 1$
$BruitPreliminaire(l) = EspaceEchantillon(i)$
end for
end for
for $k = 1$ **to** l **do**
$VecteurBruit(k) = BruitPreliminaire(T(k))$
end for
return $VecteurBruit$
	⇨ $\boldsymbol{F(i)}$ est le nombre du $i^{ème}$élément de l'espace d'échantillonnage.
	⇨ **N** est le nombre d'éléments dans l'espace d'échantillonnage.
	⇨ **T** est le vecteur Transposition

Une transposition, dite aussi permutation, est une transformation linéaire bijective qui permute les positions des éléments d'une séquence pour créer une autre séquence. Afin de définir les fonctions de transposition *T* à utiliser, le Bloc Concepteur des transpositions doit transformer une séquence de *X* nombres en une transposition avec *X* éléments. Ces nombres doivent être générés par un générateur de vrais nombres aléatoires , comme les générateurs cité dans (Danger et al. 2009; Drutarovsky and Galajda 2007; Fischer and Drutarovský 2002; Jun and Kocher 1999; Sunar et al. 2007; Tkacik 2002), ou par un générateur de nombres pseudo-aléatoires cryptographiquement sécurisé (Cryptographically-Secure Pseudo-Random Number Generator, CS-PRNG) comme le générateur Blum-Blum-Shub (BBS) (Blum et al. 1986), le CS-PRNG basé sur le Twofish(Schneier et al. 1999), le CS-PRNG Fortuna(Ferguson and Schneier 2003), le Skein (Ferguson et al. 2010), le CS-PRNG basé sur le Blowfish (Reddy and Seshadri 2013), ou autres.

L'Algorithme IV.2 représente le pseudo code de la fonction *TranspositionMaker( )*,la fonction principale du bloc Concepteur des transpositions.

Algorithme IV.2 Algorithme du bloc Concepteur des transpositions

Input : $n, graine$
Output : T
for $i = 0$ **to** n **do**
$x(i) = i$
end for
for $j = 1$ **to** n **do**
$l = 1 + floor(((n+1) - j) * NormRAND(graine))$
$T(j) = x(l)$
for $k = l + 1$ **to** $((n+1) - j)$ **do**
$x(k-1) = x(k)$
end for
end for
return T
	⇨ **n**. représente le nombre d'éléments de la transposition **T**
	⇨ **Graine** représente la graine du PRNG
	⇨ **T** est la fonction de transposition
	
	⇨ **NormRAND** est un CS-PRNG qui rend les valeurs entre 0 et 1.

Après l'ajout des $(254 - L)$ octets de bruit généré et l'octet réservé, l'Additionneur du bruit adaptatif produit les trames T-101 longues. Chaque trame T-101 longue a une longueur fixe égale à 261 octets (Figure IV.3). L'ensemble des trames T-101 longues forme une source de messages avec une distribution qui tend vers une distribution uniforme.

3. Bloc Modificateur de la structure des trames

Lors de l'analyse des en-têtes des trames IEC-101 de longueur variable et des trames longues, nous avons observé la présence d'une redondance entre le premier octet et le quatrième octet (l'octet START de valeur 0X68) d'une part, et entre le deuxième et le troisième (Les octets longueur du corps de valeur *L*) d'autre part. La suppression de ces redondances nous a permis d'introduire deux octets clés de synchronisation, ceci sans modifier la longueur de la trame ou réduire les informations des en-têtes comme le montre la Figure IV.3.

Les clés de synchronisation ont deux principales fonctions : **(1)** s'informer sur le moment de la génération des trames ; et **(2)** permettre à la sous-couche Conception des cryptogrammes de sélectionner les clés de cryptage et de décryptage afin d'implémenter un système de confidentialité

parfaite (synchronisation entre la source de clés de décryptage avec la source de clés de cryptage). En effet, si tous les nœuds de communication sont synchronisés (comme dans le cas des horloges des RTU et MTU du même système), les récepteurs peuvent connaître le moment exact de l'envoi des trames en se basant sur ces clés de synchronisation ; par conséquent, ils peuvent ignorer les trames reçues en retard. Ce processus protège le bon fonctionnement du système et empêche la réutilisation de précédentes trames pour exécuter de nouvelles instructions sous forme de cyber-attaque (Replay attacks).

Les valeurs des clés de synchronisation V_{ks}, codées sur deux octets chacune, représentent le reste de la division par 65,536 de la différence entre le moment de la réception de la trame T_{now} et le moment prédéfini T_0 :

$$V_{ks} = (T_{NOW} - T_0)\ mod\ (65536) \qquad (4.7)$$

Le moment prédéfini T_0 est déterminé par l'utilisateur. Il peut être choisi comme le moment de la première utilisation du système, du protocole ou d'une nouvelle clé. Comme il peut être le début d'un calendrier, d'un projet ou n'importe quel moment passé ou pas.

4. Format des autres trames

En plus des trames de longueurs variables, la couche Liaison de données peut concevoir d'autres types de trames : **(1)** les trames IEC-101 de longueur variable avec 261 octets, **(2)** les trames IEC-101 de longueur fixe, et **(3)** les caractères de contrôle du signal.

Lorsque la couche Liaison du protocole IEC-60870-5-101 envoie une trame IEC-101 de longueur variable avec 261 octets, la sous-couche Egalisateur de la distribution modifie la structure de son en-tête sans ajouter ni bruit ni octet de réserve. D'autre part, la couche Sécurité choisit d'une manière aléatoire et crypte des trames IEC-101 de longueur fixe et / ou des caractères de contrôle du signal afin **(1)** d'obtenir une distribution uniforme des trames envoyées et **(2)** de camoufler les trames porteuses de données SCADA (Trames IEC-101 de longueurs variables) dans le but d'empêcher les cryptanalystes de les distinguer.

Pour ces trois types particuliers de trames, le bloc additionneur du bruit adaptatif génère des trames T-101 longues par l'ajout de $(254 - L)$ octets du bruit, où $L = 0$ pour les caractères de contrôle du signal et $L \in \{1,2,3\}$. Pour les trames de longueur fixe. Puis, le Modificateur de la structure des trames modifie les en-têtes de ces trames longues par la suppression des redondances et l'ajout des

clés de synchronisation et des octets de réserve comme expliqué précédemment. La Figure IV.5 représente la structure détaillée des trames T-101 longues modifiées créées à partir des trames IEC-101 de longueur égale à 261 octets, des trames de longueur fixe et des caractères de contrôle du signal.

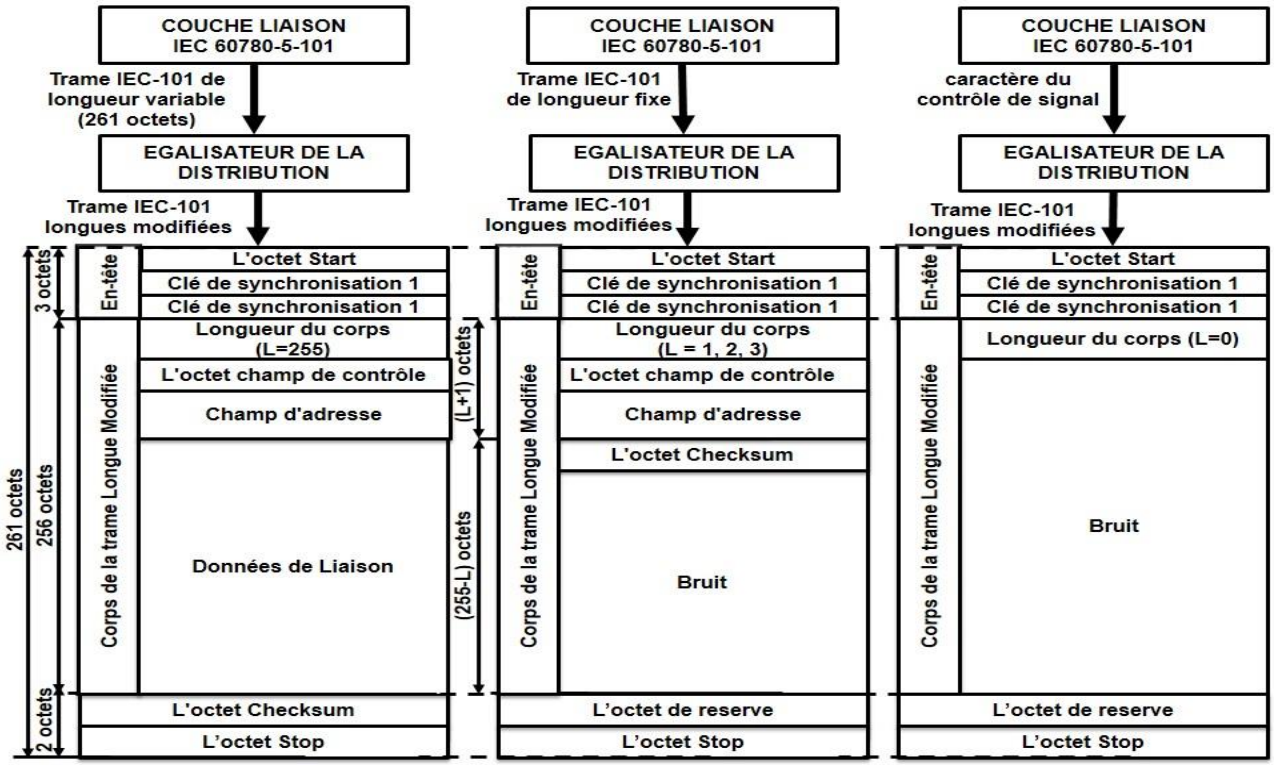


Figure IV.5 Structure des trames IEC-101 longues modifiées spéciales.

III. Sous-couche Conception des cryptogrammes

Le rôle principal de la sous-couche Conception des cryptogrammes est d'encrypter les trames T-101 longues modifiées ayant une distribution qui tend vers une distribution uniforme. Nous avons conçu cette sous-couche à partir de trois blocs XOR, qui modifient les valeurs des octets, et de trois blocs de transposition, qui modifient les positions des octets des trames. Cette conception est basée sur des opérations basiques élémentaires afin d'assurer une implémentation optimale et rapide. La sous-couche Conception des cryptogrammes utilise une nouvelle clé de cryptage pour chaque nouvelle trame afin d'implémenter la confidentialité parfaite de Shannon(Knudsen 1994; Shannon 1949). Cette implémentation assure une sécurité inconditionnelle et empêche la réutilisation des trames capturées. Elle est extrêmement efficace contre les attaques par modification et par fabrication.

1. Notions de base

La sous-couche Conception des cryptogrammes encrypte les trames T-101 longues modifiées avec une implémentation de la confidentialité parfaite basée sur le masque jetable (One Time Padcipher). Il emploie deux générateurs de clés : **(1)** Le générateur de clés du codage par XOR, et (**2**) Le générateur de clés du codage par Transposition. Périodiquement, chaque générateur produit une table de 256 clés de 256 octets chacune. La Figure IV.6 représente l'implémentation de la sous-couche Conception des cryptogrammes.

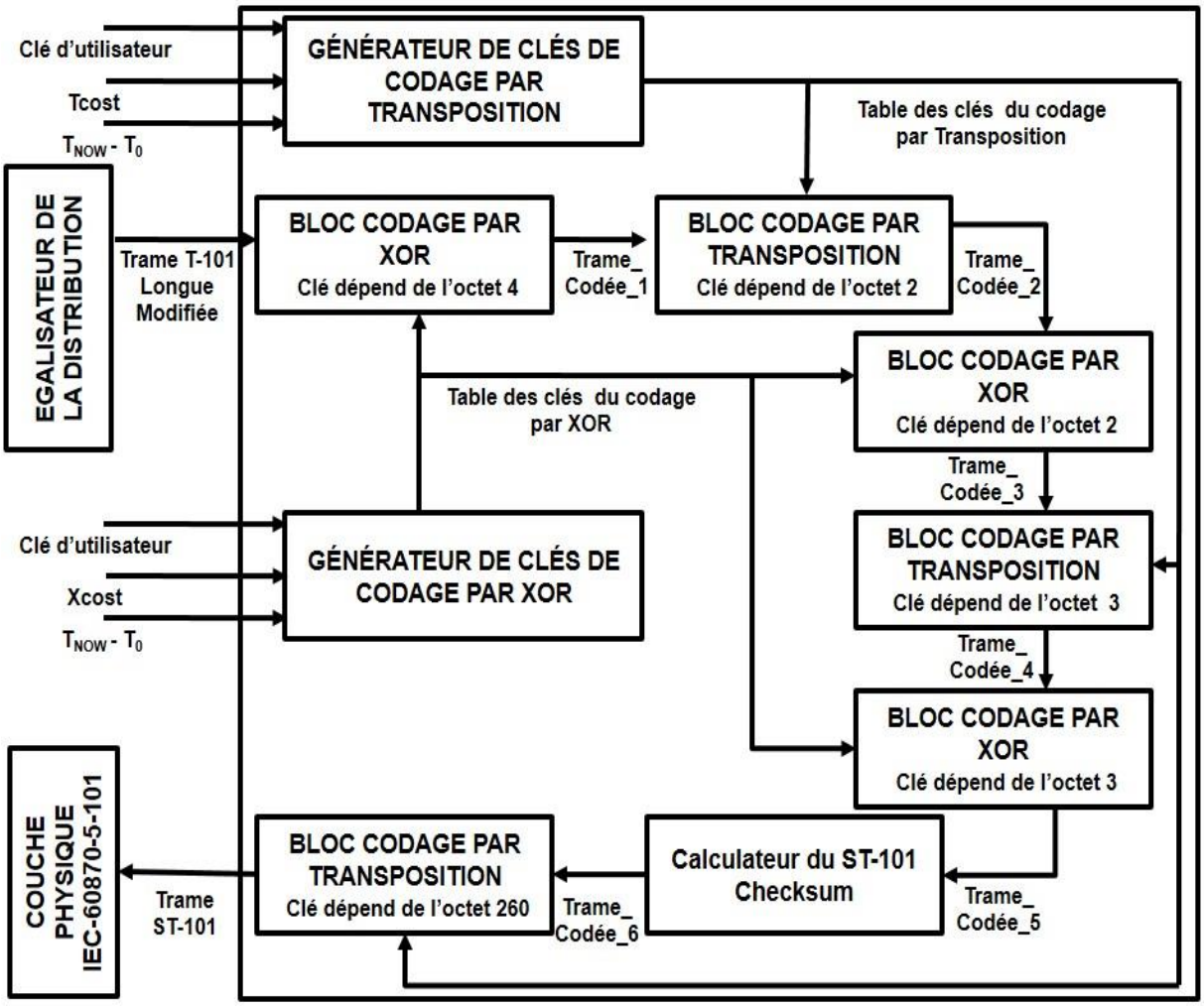


Figure IV.6 Implémentation de la sous-couche Conception des cryptogrammes

Lorsque la sous-couche Egalisateur de la distribution envoie une nouvelle trame T-101 longue modifiée, la sous-couche Conception des cryptogrammes la traite en utilisant sept blocs de codage :

Premièrement, un bloc de codage par XOR transforme la trame T-101 longue modifiée en une autre trame dite trame_codée_1. Il utilise une clé de codage XOR sélectionnée, de la table des clés du codage par XOR, en se basant sur la longueur L du corps de la trame IEC-101 originale (valeur du quatrième octet de la trame longue modifiée). Il applique une opération logique XOR entre la clé sélectionnée et la portion de 255 octets de la trame définie entre le 5ème octet et le 259ème octet.

Le second bloc est un bloc de codage par transposition. Il transforme la trame_codée_1 en une trame_codée_2 en appliquant la fonction de transposition $TranspositionEncryption(\ )$à la portion de 256 octets définie entre l'octet 4 et l'octet 259 de la trame. La clé de ~~la~~ transposition est sélectionnée, de la table des clés du codage par transposition, relativement à la valeur de la première clé de synchronisation (valeur du deuxième octet de la trame).

L'Algorithme IV.3 présente le pseudocode de la fonction $TranspositionEncryption(\ )$utilisée par les blocs du codage par transposition.

Algorithme IV.3 Fonction TranspositionEncryption ()

Input : $PartieTrame, T$
Output : $PartieCodee$
$m = Length(PartieTrame)$
for $i = 1$ ***to*** m ***do***
$PartieCodee(i) = PartieTrame(T(i))$
end for
return $PartieCodee$
	⇨ **PartieTrame**. représente la partie de la trame à encrypter ;
	⇨ **T** est le vecteur transposition sélectionné de la table ;
	⇨ **PartieCodee** est le résultat de l'encryptage.

Le troisième bloc est un bloc de codage par XOR. Il forme la trame_codée_ 3 en encryptant la trame_codée_ 2. Il applique l'opérateur XOR à la partie de la trame définie entre l'octet 4 et l'octet 259 d'une part et la clé du codage par XOR sélectionnée dépendamment de la valeur de la première clé de synchronisation.

Les quatrième et cinquième blocs ressemblent respectivement aux deuxième et troisième blocs. En effet, ils construisent les trames trame_codée_ 4 et trame_codée_ 5 en sélectionnant une clé sur la base de la valeur de la deuxième clé de synchronisation (l'octet 3 de l'en-tête), et en appliquant un

codage par transposition, suivi d'un codage par XOR sur les 256 octets des trames définies de l'octet 4 à l'octet 259.

Le sixième bloc est appelé Calculateur du ST-101 Checksum (Secure T-101 Checksum), qui calcule la somme de contrôle des 258 octets de la partie de la trame_codée_ 5 comprise entre l'octet 2 et l'octet 259. Il forme la trame_codée_ 6 par l'insertion de la valeur calculée dans l'octet 260 (c'est-à-dire dans l'octet de réserve de la trame T-101 longue modifiée).

Le dernier bloc est un bloc de codage par transposition, il réarrange les 256 octets de la trame_codée_6 définie de l'octet 4 à l'octet 259. Il utilise une clé de transposition sélectionnée selon la valeur de l'octet Checksum de la trame_codée_ 6 (l'octet 260).

Après toutes ces transformations, la trame sécurisée ST-101 sera prête à être transmise à la couche Physique pour émission.

2. Génération de clés de codage par XOR

Le générateur de clés de codage par XOR produit périodiquement une table de clés qui contient 256 clés pseudo-aléatoires de 256 octets chacune. Chaque table peut être utilisée durant 65536 unités temporelles.

Le générateur de clés de codage par XOR utilise une version **modifiée** du crypto-système Bcrypt pour assurer un haut niveau de la confidentialité et une forte résistance aux attaques passives. La force du Bcrypt réside dans son utilisation du Blowfish, le crypto-système rapide et hautement sécurisé, avec un hachage répétitif de la clé dépendant d'un poids défini par l'utilisateur(Mathur and Kesarwani 2013). Par ailleurs, le générateur de clés de codage par XOR remplace le Blowfish par la fonction de hachage sécurisée Skein. On a utilisé, au lieu de la valeur fixe du poids de Bcrypt, un poids de valeur variable générée en continu par un générateur de nombres pseudo-aléatoires cryptographiquement sécurisés SC-PRNG afin de garantir une pseudo-indépendance entre les clés générées.

L'Algorithme IV.4 présente le pseudo code de la fonction XORTableGenerator() chargé de la génération des tables de 256 clés de codage par XOR.

Algorithme IV.4 Fonction XORTableGenerator ()

$\boldsymbol{Input}: Poids, Graine$ $\boldsymbol{Output}: Poids\_Sortie, XKT$ $\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ 255\ \boldsymbol{do}$ $Poids = CSPRNG(Poids)$ $\boldsymbol{for}\ k = 1\ \boldsymbol{to}\ \boldsymbol{f}(poids)\ \boldsymbol{do}$ $Graine = Hash(Graine)$ $\boldsymbol{end\ for}$ $XKT(i) = Graine$ $\boldsymbol{for}\ j = 2\ \boldsymbol{to}\ 4\ \boldsymbol{do}$ $Poids = CSPRNG(Poids)$ $\boldsymbol{for}\ k = 1\ \boldsymbol{to}\ \boldsymbol{f}(poids)\ \boldsymbol{do}$ $Graine = Hash(Graine)$ $\boldsymbol{end\ for}$ $XKT(i) = XKT(i) + Graine$ $\boldsymbol{end\ for}$ $\boldsymbol{end\ for}$ $Poids\_Sortie = Poids$ $\boldsymbol{return}\ Poids\_Sortie, XKT$	⇨ **Poids**. représente le poids de CS-PRNG ; ⇨ **Graine** est la graine de la fonction de hachage ; ⇨ **Graine_Sortie** est la dernière graine de hachage, utilisée pour générer la prochaine table ; ⇨ **XKT** est la table des clés de codage par XOR ⇨ **CSPRNG()** est la fonction du générateur de nombres pseudo-aléatoires cryptographiquement sécurisé ⇨ ***f()*** est la fonction de modification du temps d'exécution ; ⇨ **Hash()** est la fonction de hachage cryptographique de 512 bits

La fonction XORTableGenerator() doit exploiter une fonction de hachage rapide et sécurisée de 64 octets en sortie. Pour cette raison, nous avons choisi la fonction de hachage Skein basée sur le crypto-système Threefish. Il est conçu pour être rapide, simple, flexible et hautement sécurisé (Ferguson et al. 2010; Kong et al. 2015). Son architecture modulaire, basée sur les blocs d'addition, de rotation et XOR ARX sécurisé, l'a classé parmi les cinq finalistes de la compétition SHA-3 (Ferguson et al. 2010). Il résiste aux attaques de type collision les plus connues, aux attaques différentielles et de récupération de clé impossibles, aux attaques de pré-image et de pseudo-pré-image, aux attaques de canal latéral et les attaques de rebond rotatif (Bellare et al. 2009; Chang et al. 2012; Khovratovich et al. 2014). L'architecture ARX de la fonction de hachage Skein lui confère également d'excellentes performances logicielles, meilleures que SHA-2 ainsi que l'éventuel gagnant de SHA-3 (Chang et al. 2012; Lee et al. 2016).

La fonction XORTableGenerator() a besoin aussi d'un générateur de nombre pseudo-aléatoire cryptographiquement sécurisé. Le CS-PRNG Blum-Blum-Shub, appelé aussi le générateur BBS $x^2 mod(N)$, semble être un excellent choix. Basé sur la difficulté du Problème de la résiduosité quadratique, Le BBS est connu comme l'un des CS-PRNG les plus sécurisés et les plus efficaces (Blum et al. 1986; Gennaro 2000; Sidorenko and Schoenmakers 2005).

Finalement, le temps d'exécution de la fonction XORTableGenerator() est augmenté intentionnellement par l'usage de la fonction de modification du temps d'exécution *f*(). En effet, augmenter le temps d'exécution de la fonction empêche les cyber-attaquants d'effectuer un nombre suffisant d'essais permettant de deviner la clé utilisée. La fonction de hachage Bcrypt utilise la fonction exponentielle à base de 2 comme fonction de modification du temps d'exécution (Provos and Mazieres 1999). Afin d'augmenter plus la complexité de l'algorithme, nous avons proposé d'utiliser des valeurs variables pseudo-aléatoires comme poids d'algorithme pour assurer une pseudo-indépendance importante entre les clés générées, en plus de celle assurée par l'utilisation des algorithmes BBS et Skein.

L'Algorithme IV.5 présente le pseudo code de la fonction K-XORTableGenerator() chargé de la génération de la K^{ième} table de clés de codage par XOR. Où *K* est donné par la division entière :

$$K = \left(\frac{T_{now} - T_0}{65536}\right) + 1 \qquad (4.8)$$

Algorithme IV.5 Fonction K-XORTableGenerator ()

$\boldsymbol{Input}: Xpwd, Xpoids, K$ $\boldsymbol{Output}: Poids\_Sortie, XKT$ $Graine = Xpwd$ $Poids = XPoids$ $\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ K\ \boldsymbol{do}$ $[Poids, XKT]$ $= XORTableGenerator(Poids, Graine)$ $Graine = XKT(255)$ $\boldsymbol{end\ for}$ $Poids\_Sortie = Poids$ $\boldsymbol{return}\ Poids\_Sortie, XKT$	⇨ **Xpwd** est la clé de l'utilisateur ; ⇨ **Xpoids** est le poids proposé par l'utilisateur ; ⇨ **K** est l'ordre de la table des clés ; ⇨ **Poids_Sortie** est le dernier poids calculé ; ⇨ **XKT** est la table de clés de codage par XOR ;

3. Génération de clés de codage par transposition

Le générateur de clés de codage par transposition produit des tables de 256 vecteurs (clés) de transposition, où chaque table peut être utilisée pour 65536 unités temporelles. Il applique le même principe que le générateur de clés de codage par XOR.

L'Algorithme IV.6 présente le pseudo code de la fonction TranspositionTableGenerator() utilisée par le générateur de clés de codage par transposition pour produire les tables de clés.

Algorithme IV.6 Fonction TranspositionTableGenerator ()

Input : $Poids, Graine$
Output : $Poids\_Sortie, TKT$
 	
$Poids = CSPRNG(Poids)$
for $i = 1$ ***to*** $f(poids)$ ***do***
  $Graine = Hash(Graine)$
end for
$PrGraine = Graine$
for $i = 2$ ***to*** 4 ***do***
  $Poids = CSPRNG(Poids)$
for $j = 1$ ***to*** $f(poids)$ ***do***
  $Graine = Hash(Graine)$
  end for
  $PrGraine = PrGraine + Graine$
end for
$TKT(1)$
$= TranspositionMaker(PrGraine)$
for $i = 2$ ***to*** $\mathbf{256}$ ***do***
  $TKT(i) = TranspositionMaker(TKT(i - 1))$
end for
$Poids\_Sortie = Poids$
return $Poids\_Sortie, TKT$
	⇨ **Poids**. représente le poids de CS-PRNG ;
	⇨ **Graine** est la graine de la fonction de hachage ;
	⇨ **Graine_Sortie** est la dernière graine de hachage, utilisée pour générer la prochaine table ;
	⇨ **TKT** est la table des clés de codage par transposition ;
	⇨ **CSPRNG()** est la fonction du générateur de nombres pseudo-aléatoires cryptographiquement sécurisée ;
	⇨ ***f()*** est la fonction de modification du temps d'exécution ;
	⇨ **Hash()** est la fonction de hachage cryptographique de 512 bits

Pour chaque table, l'Algorithme produit une graine primaire *PrGraine* composée de quatre parties : la première partie est générée par le hachage de la graine initiale $f(CSPRNG(Poids))$ fois. Les trois

autres parties sont générées à partir de la partie précédente en utilisant le même processus. Il est à noter que *f*() est la fonction de modification du temps d'exécution. Le générateur de nombres pseudo-aléatoires choisi est le générateur de Blum-Blum-Shub, alors que la fonction de hachage sélectionnée est la fonction Skein.

Ensuite, la fonction *TranspositionMaker*(), basée sur le générateur BBS normalisé, est appliquée sur la *PrGraine* pour générer *TKT*(1) la première clé de codage par transposition de la table. Afin de générer une autre clé $TKT(i)$ de la table à partir des autres clés, on applique la fonction *TranspositionMaker*(), basée sur le générateur BBS normalisé et la clé précédente $TKT(i-1)$.

$$TKT(i) = TranspositionMaker(TKT(i-1)) \qquad (4.10)$$

Finalement, la fonction TranspositionTableGenerator() renvoie la table de clés TKT pour qu'elle soit utilisée par les blocs de codage par transposition. La dernière clé de la table TKT(256) et le Poids final sont généralement utilisés comme Graine et Poids de la prochaine table.

IV. Décryptage des trames ST-101

La couche Sécurité entame une procédure de décryptage des trames ST-101 une fois qu'elle les reçoit. La Figure IV.7 représente le diagramme de décryptage adopté par la couche.

Premièrement, l'authentification de la trame ST-101 reçue est vérifiée par le calcul de la somme de contrôle des 258 octets de la partie de la trame comprise entre l'Octet 2 et l'Octet 259. Le checksum calculé doit être identique à la valeur sauvegardée dans l'Octet 260. Dans le cas contraire, la trame serait ignorée.

Ensuite, la Trame_décodee_1 est obtenue après l'application de la fonction de décryptage des transpositions, *TranspositionDycryption*(), sur la partie de 256 octets de la trame ST-101 définis entre l'Octet 4 et l'Octet 259. La clé utilisée est sélectionnée de la table de clés de codage par transposition courante en se basant sur l'Octet 260 de la trame.

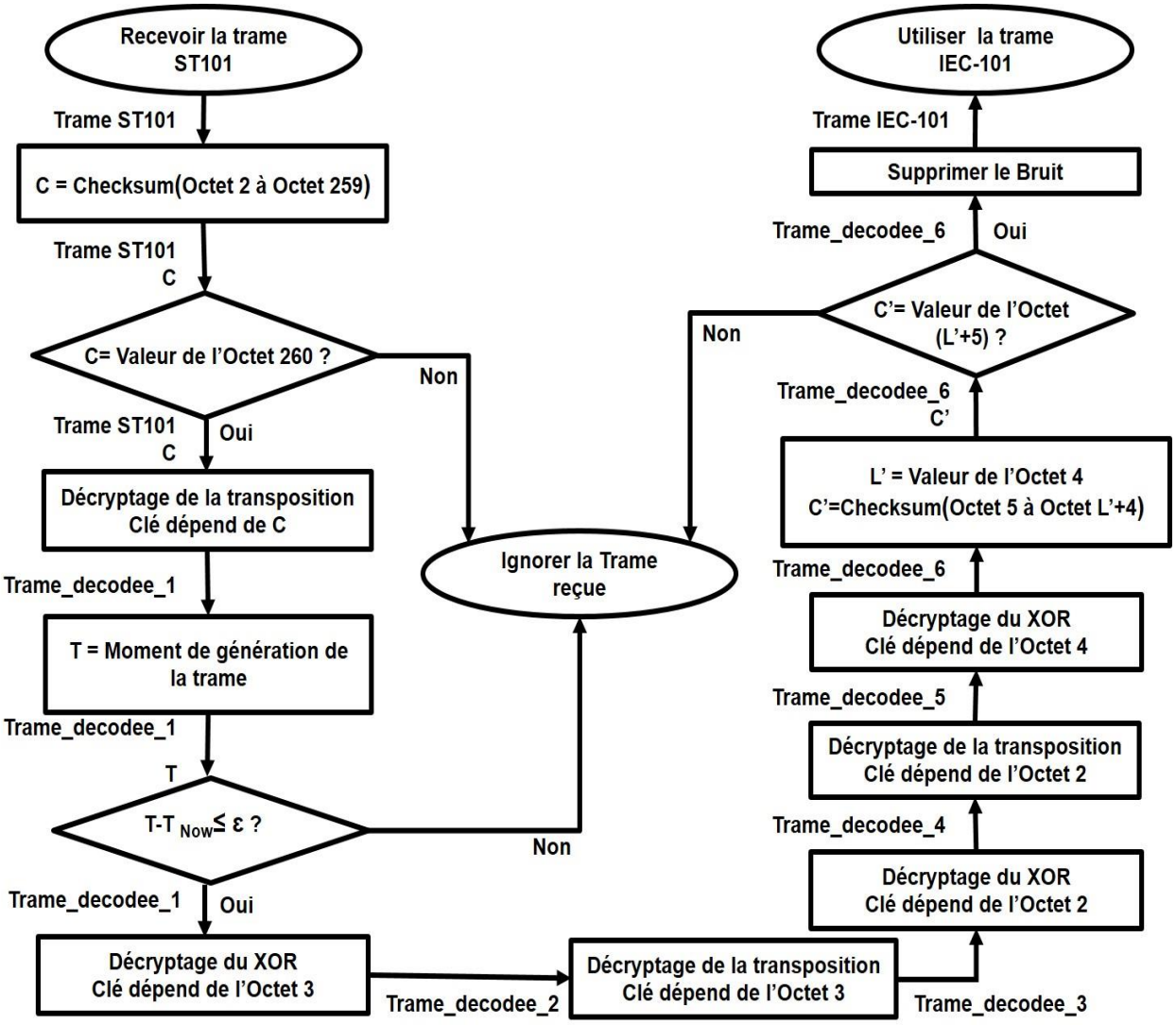


Figure IV.7 Diagramme de la procèdure de décryptage.

L'Algorithme IV.7 présente le pseudo code la fonction *TranspositionDecryption*() qui décrypte les trames encryptées par transposition.

Algorithme IV.7 Fonction TranspositionDecryption()

Input : $Partie\_Codee, T$
Output : $Partie\_Decodee$
 	
for $i = 1$ ***to*** m ***do***
$Partie\_Decodee(T(i)) = Partie\_Codee(i)$
end for
return $Partie\_Decodee$
	⇨ **Partie_Codee** est la partie de la trame à décrypter;
	⇨ **T** est la fonction de transposition ;
	⇨ **Partie_Decodee** est le résultat du décryptage de Partie_Codee ;
	⇨ **m** est la taille de Partie_Codee.

Les deuxième et troisième octets de la Trame_Decodee_1 correspondent aux octets clés de synchronisation. Si la différence entre le moment de la réception de la trame et son moment de génération, indiqués par ces deux octets, est supérieure au temps acceptable de la transmission, alors la couche Sécurité va arrêter le décryptage et ignorer la trame reçue en la considérant comme issue d'une cyber-attaque.

Si la trame reçue est arrivée à temps, la couche Sécurité doit transformer la Trame_decodee_1 en une Trame_decodee_2 par l'application d'un décryptage par XOR à la portion de 256 octets, déterminée entre l'Octet 4 et l'Octet 259. Elle utilise, pour cette opération, une clé de la table de clés de codage par XOR sélectionnée conformément à la valeur de l'Octet 3 de la trame. Puis, on obtient la Trame_decodee_3 par un décryptage de transposition basé sur la valeur de l'Octet 3 et appliqué sur les 256 octets de la Trame_decodee_2 située entre l'Octet 4 et l'Octet 259.

Après cette étape, un autre décryptage par l'opérateur XOR est appliqué sur les 256 octets de la Trame_decodee_3, de l'Octet 4 à l'Octet 259. La clé utilisée, cette fois-ci, est choisie dépendamment de la valeur de l'Octet 2 de la trame. On obtient ainsi laTrame_decodee_4. De même, on construit la Trame_decodee_5 par l'application de la fonction *TranspositionDecryption*() sur la même partie et en choisissant une clé de la table des clés de codage par transposition conformément à la valeur de l'Octet 2 de la trame.

Dans l'étape de décryptage suivante, la couche Sécurité applique un décryptage par XOR sur les 255 octets définis entre l'Octet 5 et l'Octet 259, la clé utilisée dépend de la valeur de l'Octet 5 de la Trame_decodee_5. La trame résultante de cette étape est la Trame_decodee_6 qui doit correspondre à la trame T-101 longue modifiée envoyée.

Pour vérifier l'intégrité de la trame reçue, la valeur de l'Octet $(5 + L')$ doit être égale à la valeur du checksum des L' octets de la Trame_decodee_6 compris entre l'Octet 5 et l'Octet$(4 + L')$, où L' est la valeur de l'Octet 4 de la Trame_decodee_6. Si le checksum calculé est différent de la valeur de l'Octet $(5 + L')$, alors la couche Sécurité doit arrêter le décryptage et ignorer la trame reçue.

Si l'intégrité de la Trame_codee_6 est vérifiée, la trame standard IEC-101 peut être construite en se basant sur la valeur de l'Octet 4 de la trame. Cet octet correspond à la longueur de la trame standard L :

Si $L > 3$, La trame T-101 Standard est une trame de longueur variable. Le corps de la trame d'origine est composé de la partie de la Trame_decodee_6 comprise entre l'Octet 5 et l'Octet $(4 + L)$. L'Octet $(5 + L)$ contient la valeur de la somme de contrôle de la trame T-101standard.

Si $L \in \{1,2,3\}$, La trame T-101 standard est une trame de longueur fixe. Son corps correspond à la partie de la Trame_decodee_6 entre l'Octet 5 et l'Octet $(4 + L)$. Le checksum de la trame IEC-101 standard est donné par l'Octet $(5 + L)$ de la Trame_decodee_6. Pour $L = 0$, la trame IEC-101 envoyée ne contient que le caractère de contrôle du signal.

Finalement, l'émetteur peut être authentifié en vérifiant la valeur de son adresse exprimée dans le champ d'adresse.

V. Satisfaction des contraintes temporelles

La satisfaction des contraintes temporelles représente une condition nécessaire pour l'adoption de toute solution SCADA. En effet, les systèmes SCADA, utilisés par les infrastructures critiques, sont des systèmes temps réel qui subissent des contraintes temporelles critiques. L'habileté de notre solution à s'introduire dans la gestion des infrastructures électriques a été évaluée par l'analyse des réponses temporelles d'une implémentation de la couche Sécurité sur un ordinateur industriel. Ce dernier est caractérisé par son processeur CPU Intel I5-5200U avec Dual cores de 2.2GHz et un turbo boost de 2.7 GHz, une mémoire vive DDR3 SDRAM de 4Go, un disque dur mSATA SSD de 64 GHz, quatre ports RS232 et un port Gigabit Ethernet.

Premièrement, nous avons évalué la durée nécessaire, à l'additionneur du bruit, le modificateur des trames, les trois blocs codeurs par XOR, les trois blocs codeurs par transposition et le bloc calculateur du ST-101 checksum, pour transformer un million de trames IEC-101de longueurs variables successives à des trames ST-101 longues (et vice versa). Dans notre évaluation, les vecteurs de bruits et les tables de clés ont été générés indépendamment du processus de cryptage et de décryptage des trames.

La Figure IV.8 présente la distribution de nombres de trames IEC-101 de longueurs variables successives encryptées en fonction de la durée mesurée d'encryptage. Et la Figure IV.9 présente la distribution de nombres de trames ST-101 longues en fonction de la durée mesurée de décryptage.

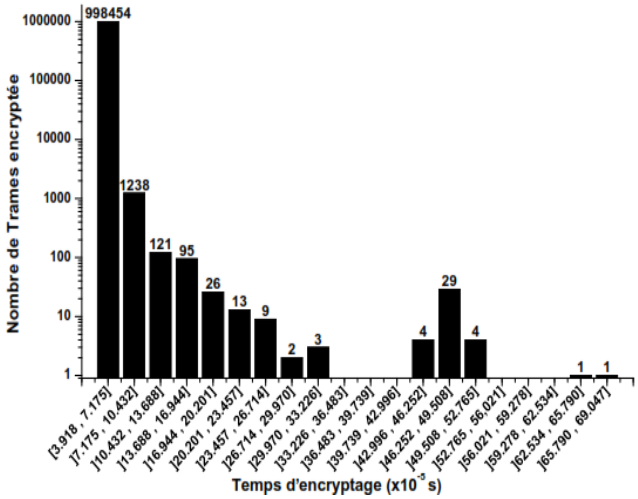


Figure IV.8 Distribution de nombres de trames IEC-101 successives encryptées en fonction de la durée d'encryptage.

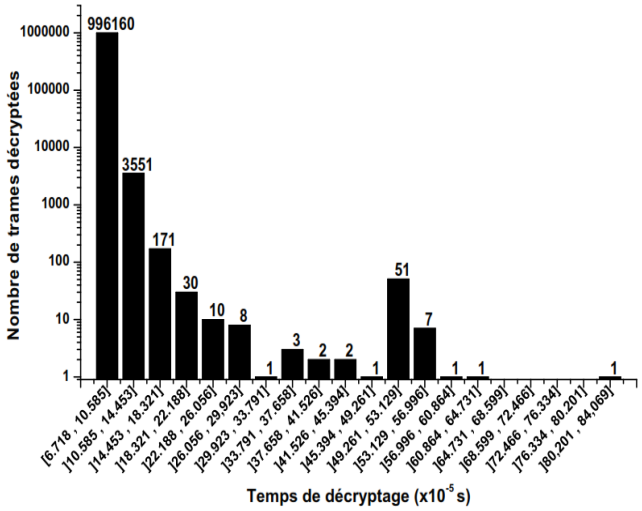


Figure IV.9 Distribution de nombres de trames ST-101 longues modifiées en fonction de la durée de décryptage.

L'analyse de ces résultats révèle que la durée moyenne mesurée d'encryptage des trames IEC-101 de longueurs variables est de 43.124×10^{-6} secondes avec un écart type de 5.1442×10^{-6} secondes et une durée maximale d'encryptage de 69.047×10^{-6} secondes. Alors que la durée moyenne de décryptage des trames ST-101 longues est de 74.724×10^{-6} secondes avec un écart type de 7.0903×10^{-6} secondes et une durée maximale de décryptage de 84.069×10^{-6} secondes. Ces durées sont négligeables comparées à la durée théorique nécessaire à l'envoi d'une trame IEC-101 de 261 octets (cette durée est égale à 299.06×10^{-3} seconde avec la configuration du port RS-232 : huit bits de données, un bit parité et un bit stop).

Ensuite, nous avons mené des expériences pour évaluer le temps de réponse de la mise en œuvre du protocole ST-101 sous un système SCADA et la satisfaction des contraintes temporelles liées à la conduite des réseaux d'électricité. Le temps de réponse est défini comme un intervalle de temps s'étendant du lancement de la requête d'interrogation à la réception de la réponse. L'évaluation a été réalisée en comparant les temps de réponse du système SCADA ST 101 avec les temps de réponse des systèmes SCADA IEC-101 existants utilisés pour gérer les postes sources du réseau d'électricité.

Nous avons mesuré et analysé les réponses temporelles de trois types de stations terminales éloignées RTU. La RTU_01 fonctionne avec le protocole ST-101 sous l'ordinateur industriel décrit précédemment. Les deux autres RTU sont des RTU à usage répandu dans la téléconduite des réseaux électriques : La RTU_02 qui fonctionne avec le protocole IEC-101 sous un ordinateur

industriel avec un processeur CPU Intel Pentium IV de 2.4GHz, une mémoire vive SDRAM DDR2 de 2Go, un disque dur HDD de 80Go, deux ports RS-232 COM, un port RJ-45 Fast Ethernet. Et finalement, la RTU_03 ayant une architecture modulaire et dont le module de communication chargé du protocole de communication IEC-101 a un processeur CPU Intel 80486 de 66MHz, une mémoire vive de 4 Mo, une mémoire Flash de 4 Mo et quatre ports série RS-232 COM.

Notre système expérimental comporte les trois RTU connectées à un réseau de terrain, identique à ceux exploités par les postes sources électriques opérationnels. Chaque RTU utilise le protocole de communication niveau bas Modbus RTU pour gérer 2308 adresses d'objets. Leurs ports séries RS-232 sont paramétrés avec la configuration recommandée : huit bits de données, une parité paire et un bit stop. La Figure IV.10 représente l'architecture du système expérimental utilisé pour l'évaluation du temps de réponse.

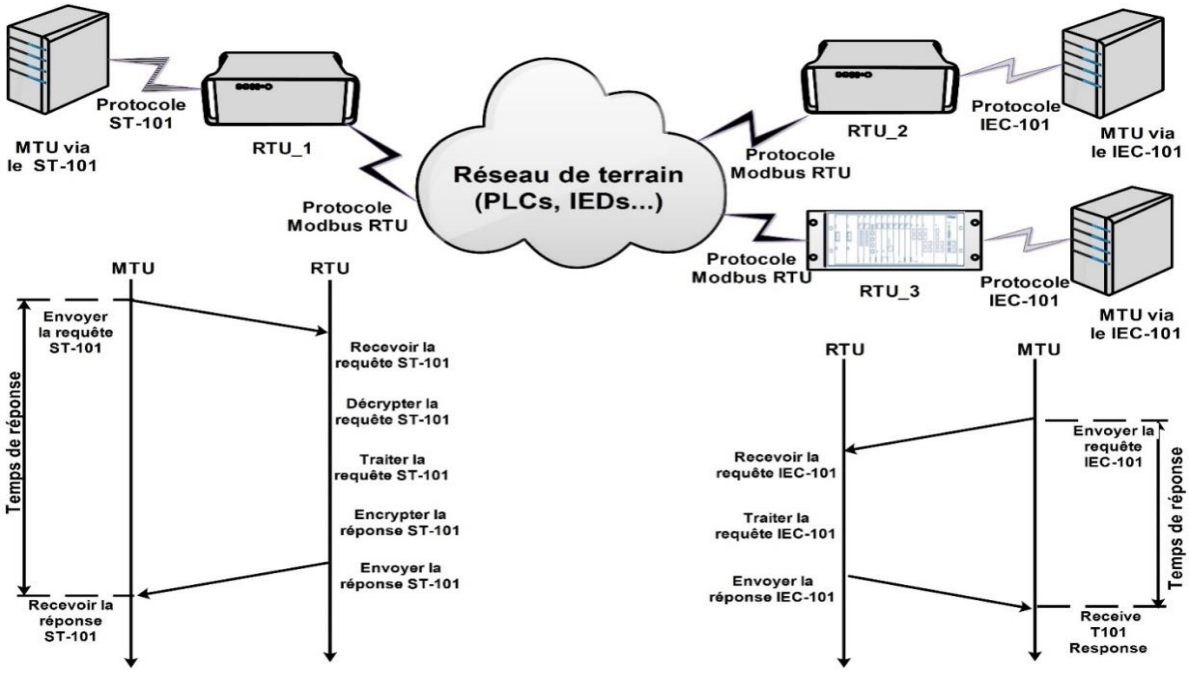


Figure IV.10 Système expérimental d'évaluation de la réponse temporelle

Les tests expérimentaux révèlent que la réponse temporelle de la RTU_01 est comparable à celle de la RTU_02 et meilleure que celle de la RTU_03. En effet, la RTU_01 répond aux requêtes ST-101 dans une durée de réponse moyenne de 422.32×10^{-3} secondes avec un écart type de 7.834×10^{-3} secondes. La RTU_02 répond aux requêtes IEC-101 avec des trames longues (de longueurs supérieures à 250 octets) dans une durée de réponse moyenne de 404.27×10^{-3}secondes avec un écart type de 11.03×10^{-3} secondes. La dernière RTU, la RTU_03, répond aux requêtes IEC-101

avec des trames longues dans une durée moyenne de 579.60×10^{-3} secondes avec un écart type de 14.73×10^{-3}secondes.

La comparaison expérimentale de l'implémentation confirme que notre solution proposée basée sur le protocole sécurisé ST-101 satisfait les contraintes temporelles imposées par les systèmes SCADA utilisés pour la téléconduite des réseaux électriques. Elle n'affecte pas le fonctionnement temps-réel des opérations de téléconduite, spécifiquement pour des implémentations mieux optimisées.

VI. Discussions

Nous avons présenté dans ce chapitre notre nouvelle approche développée pour sécuriser les communications SCADA temps réel. L'objectif de cette approche est de sécuriser le protocole IEC 60870-5-101 par l'introduction d'une couche Sécurité entre la couche physique et la couche liaison de son architecture à performance avancée EPA. Son rôle est de garantir un niveau haut de confidentialité conformément à la notion de sécurité inconditionnelle de Shannon. Elle représente une implémentation pratique du principe de la confidentialité parfaite et du principe de la confidentialité idéale forte.

La couche Sécurité incorpore deux sous-couches : La sous-couche Egalisateur de la distribution et la sous-couche Conception des cryptogrammes. La sous-couche Egalisateur de la distribution implémente le principe de la confidentialité idéale forte par la modification des caractéristiques probabilistes des messages envoyés. Elle est conçue pour satisfaire la notion du déterminisme temporel définie par le support d'une émission continue de trames IEC 60870-5-101 de 261 octets de longueur. La sous-couche ajoute un bruit pseudo-aléatoire aux trames IEC-101 de longueurs variables, certaines trames de longueur fixe et d'autres caractères de contrôle du signal pour créer les trames T-101 longues modifiées de 261 octets. Pour une distribution de bruit ajouté conforme aux relations (4.2) et (4.6), la distribution des messages envoyés tend à être uniforme. Dans cette situation, les communications sont sécurisées indépendamment du cryptage utilisé.

Le traitement de certaines trames de longueurs fixes et des caractères de contrôle du signal par la couche de Sécurité a deux objectifs. Le premier est de permettre au système d'introduire suffisamment de bruit aux trames pour avoir une distribution uniforme. Le second objectif est de les utiliser pour camoufler les trames de longueurs variables porteuses de données.

Après la modification des caractéristiques probabilistes des messages à transmettre, la sous-couche Egalisateur de la distribution change les en-têtes de trames T-101 longues modifiées par l'introduction des deux octets Clés de synchronisation. Ces deux octets marquent le moment de la construction des trames. Ils permettent, ainsi, la sélection des clés de cryptage et empêchent la réutilisation des trames déjà envoyées pour construire une cyber-attaque.

La sous-couche Conception des cryptogrammes est une implémentation pratique du principe de la confidentialité parfaite. Elle alterne entre trois blocs de cryptage par XOR et trois autres blocs de cryptage par transposition. Chaque bloc de cryptage sélectionne la clé de cryptage à partir d'une table de clés en se basant sur (1) la longueur originale de la trame IEC-101 à crypter, (2) les valeurs des octets de synchronisation, et (3) de la valeur du checksum de la trame ST-101 créée comme le montre la Figure IV.6.

La sous-couche Conception des cryptogrammes génère périodiquement de nouvelles tables de clés de codage par XOR et de codage par transposition en se basant sur (1) la clé de l'utilisateur, (2) le poids, (3) le moment de l'envoi de la trame et (5) un moment prédéfini.

Pour obtenir une forte pseudo-indépendance et un haut niveau de confidentialité des clés des tables, nous avons choisi d'utiliser le CS-PRNG Blum-Blum-Shub, le générateur de nombre pseudo-aléatoire hautement sécurisé, et l'algorithme de hachage sécurisé Skein (le rapide, le simple, le sécurisé, le flexible et le fiable finaliste du concours SHA-3), pour générer les tables de clés. La sous-couche utilise, aussi, une fonction de modification du temps d'exécution pour augmenter, intentionnellement, la complexité et le temps de la génération des clés comme il est spécifié par l'Algorithme IV.4 et l'Algorithme IV.6. Cette implémentation assure la vérification de l'authenticité du système et le protège contre les attaques passives et les attaques par modification et par fabrication.

Avec l'utilisation des deux clés de synchronisation, chaque table de clés peut être utilisée durant 65536 unités temporelles. Pour une unité temporelle égale à une seconde, chaque table de clés peut être utilisée durant 18 heures, 12 minutes et 16 secondes. Dans ce cas et avec un débit de transmission de $9600\ bits/seconde$, chaque valeur des octets des clés de synchronisation peut être utilisée pour crypter quatre trames IEC-101 au maximum. L'expérience a montré que c'est vraiment rare d'obtenir deux trames cryptées avec la même clé. La première raison de ce résultat est la non-commutativité entre les différents blocs de codage (le codage par XOR et le codage par

transposition). La deuxième raison est la dépendance des clés de cryptage aux clés de synchronisation, aux longueurs de trames IEC-101 originales et aux valeurs des checksums des trames ST-101 qui dépendent du bruit ajouté généré par un CS-PRNG. Ces deux raisons sont les raisons pour lesquelles la sous-couche Conception des cryptogrammes est considérée comme une implémentation pratique du principe de la confidentialité parfaite.

La génération des clés de cryptage nécessite l'utilisation de graines de codage par XOR et de graines de codage par transposition. Comme le partage de ces graines entre les différents équipements du système SCADA présente de sérieux risques sécuritaires, nous proposons trois mécanismes de partage de graines :

Le premier mécanisme est basé sur le partage manuel des clés. Ce mécanisme est pratique pour une première configuration des équipements ou dans le cas d'un système de nombre limité d'équipements distribués sur une zone géographiquement réduite. Dans le cas d'autres scénarios, ce mécanisme pose plusieurs problèmes techniques et logistiques.

Le deuxième mécanisme emploie un des algorithmes de cryptage sécurisés communs utilisés par les systèmes de technologies d'information. En effet, la transmission des graines ne nécessite pas un strict respect de la notion du temps réel. Par conséquent, les graines peuvent être protégées et transmises de la même manière que celle utilisée pour les données des systèmes d'information.

Le troisième mécanisme consiste à utiliser des messages de configuration sécurisée, avec notre approche proposée, pour envoyer les nouvelles graines de codage. Ces messages de configuration doivent contenir les paramètres cryptographiques tels que les nouvelles graines de codage par XOR et de codage par transposition, les graines de configuration indépendantes des autres graines, le moment prédéfini T_0, les poids de codage par XOR et par transposition ainsi que le poids de configuration.

La couche Sécurité est conçue avec une architecture modulaire pour faciliter les implémentations matérielles et parallèles. Cette conception assure la rapidité des opérations de cryptage. La création des vecteurs du bruit et la génération des tables de clés sont des processus relativement lents, mais ceci n'affecte pas le fonctionnement temps réel du système. En effet, les mêmes vecteurs du bruit et tables de clés peuvent être réutilisés pour crypter les différentes trames. Par contre, l'implémentation du protocole ST-101 sur un ordinateur industriel a montré que les opérations du traitement, tels que

l'ajout du bruit aux trames, la modification des en-têtes et les transformations de codage par XOR et par transposition, sont des opérations suffisamment rapides pour satisfaire les contraintes temporelles imposées par les systèmes SCADA.

D'un autre point de vue, cette conception peut porter un trafic continu de trames longues. Il est clair que l'augmentation du trafic accroît la vulnérabilité des systèmes SCADA aux attaques DOS. Plusieurs solutions possibles peuvent être envisagées telles que (1) l'augmentation du débit de communication lorsque la capacité du canal l'autorise, (2) réduire le trafic par le partage des adresses objets sur un nombre supérieur de stations éloignées RTU et de canaux de communication, et (3) de définir des trames ST-101 avec une taille optimale inférieure à 261 octets. Des travaux de recherche futurs sont envisagés pour déterminer la relation entre la capacité du canal, la taille des trames et les attaques DOS.

L'implémentation de la confidentialité parfaite et de la confidentialité idéale forte offre un haut niveau de confidentialité et supporte une authentification fiable. Uniquement les entités du système synchronisées peuvent utiliser des clés correctes pour encrypter les trames envoyées et décrypter les trames reçues. Vérifier l'intégrité des trames reçues et ignorer les messages avec erreurs, malformations ou mauvaises formations, comme le montre la Figure IV.7, permet de détecter et neutraliser des attaques par modification et fabrication (telles que les man-in-the-middle-attacks et les false-server-attacks).

Notre conception utilise une méthode d'étiquetage temporel pour combattre les replay-attacks. Les clés de synchronisation marquent les moments de conception des trames. A la réception des trames, le récepteur compare les moments marqués par les clés de synchronisation avec les moments de réceptions des trames. Dans le cas d'une replay-attack, la différence entre ces deux moments va être significative, le système doit simplement ignorer les trames reçues et envoyer une alerte.

Pour augmenter le niveau de confidentialité et combattre les brute-force-search-attacks et rainbow-attacks, les clés introduites par les utilisateurs ne doivent jamais être exploitées directement. Nous exigeons de les traiter avec les algorithmes IV.4 et IV.6 avant leur utilisation. Les modificateurs du temps d'exécution sont utilisés pour augmenter la complexité et le temps d'exécution du processus de génération des tables de clés. Nous avons préféré que les modificateurs du temps exploitent des nombres pseudo-aléatoires, comme poids d'algorithmes, pour augmenter la complexité et la pseudo-indépendance des clés. Pour cette raison, nous avons choisi d'employer le CS-PRNG Blum-Blum-

Shub et l'algorithme de hachage Skein. Finalement, les paramètres de configuration et les tables de clés doivent être modifiés assez fréquemment pour implémenter le one-time-pad cipher.

VII Conclusion

Ce travail présente notre protocole de transmission SCADA le T-101 sécurisé (ST-101) basé sur le protocole populaire IEC-60870-5-101. Les systèmes SCADA sont souvent utilisés pour gérer les infrastructures critiques, ce qui fait que les traditionnels mécanismes de sécurité sont inadéquats pour les sécuriser face aux cyber-attaques. Le protocole ST-101 introduit une nouvelle couche Sécurité entre la couche liaison et la couche physique de l'architecture à performance avancée du IEC-101 afin de sécuriser les messages échangés. La couche Sécurité correspond à une implémentation de la sécurité inconditionnelle de Shannon, où les principes de la confidentialité parfaite et la confidentialité idéale forte sont mis à profit pour garantir l'authenticité, l'intégrité, et la confidentialité des communications SCADA.

Le protocole ST-101 fournit un haut niveau de sécurité par l'usage de plusieurs mécanismes tels que : (1)Ajouter suffisamment de bruit pseudo-aléatoire pour garantir une distribution uniforme des messages transmis,(2)coder certaines trames de longueur fixe et de caractères de contrôle du signal pour camoufler les trames de longueurs variables porteuses d'importantes données SCADA, (3)marquer le moment de fabrication des messages envoyés pour se protéger contre la réutilisation des trames,(4)produire des clés de codage pseudo-indépendantes garanties par l'usage de forts CS-PRNG et fonctions de hachage, (5)utiliser une fonction de modification du temps d'exécution, (6) sélectionner des clés de cryptage dépendamment d'un ensemble de paramètres indépendants, (7)changer souvent les clés de cryptage, et finalement (8)ignorer les messages qui présentent des problèmes lors du décryptage.

CONCLUSION GENERALE

Au jour d'aujourd'hui, la quasi-totalité des infrastructures critiques modernes font référence aux réseaux de contrôle industriels pour gérer leurs installations techniques. Les systèmes de contrôle et d'acquisition des données SCADA sont des réseaux de contrôle industriels temps réel utilisés pour la gestion instantanée des processus industriels des installations techniques distribués sur une large zone géographique. Du point de vue architecturel, les systèmes SCADA adoptent une architecture centralisée composée de trois principaux segments : (1) Le segment principal du réseau SCADA installé au niveau du centre de contrôle, (2) Le segment des équipements de terrain regroupant tous les équipements installés aux niveaux des nœuds finaux, et (3) le segment de réseau de coopération basé sur l'ensemble des technologies de télécommunication employées pour assurer la liaison entre les deux segments précédents.

Du point de vue logiciel, l'analyse de l'architecture protocolaire de la pyramide CIM montre que les protocoles SCADA sont divisés en trois niveaux : (1) Les protocoles basiques assurant la communication des automates avec les capteurs et les actionneurs, (2) les protocoles de terrains utilisés pour la communication entre automates et les RTU, et finalement (3) les protocoles de transmission de données pour la communication de niveau haut entre les RTU et les MTU. Contrairement aux deux premières classes de protocoles SCADA, les protocoles de transmission de données transportent des volumes de données importants sur des réseaux de type WAN. Par conséquent, ils doivent fournir des mécanismes de communication simples, robustes, riches, fiables, sûrs et mis à jour.

A partir de leur troisième génération, les systèmes SCADA ont commencé à utiliser des réseaux de communication publics et des protocoles de transmission ouverts. Ceci a énormément amélioré leurs déploiements, leurs évolutions, leurs exploitations et leurs développements. Mais ceci n'a été sans aucune conséquence. L'ouverture des réseaux SCADA a nettement affaibli l'aspect sécuritaire des systèmes. Ils sont devenus plus vulnérables aux cyber-attaques externes qui s'introduisent, généralement en exploitant les failles sécuritaires des réseaux de coopération afin d'avoir un impact d'effet plus large. L'analyse de ces attaques nous a permis de les diviser en attaques passives qui visent la confidentialité du système, attaques par déni de service qui visent la disponibilité du

système et attaques par modification et fabrication des trames qui visent l'intégrité et l'authenticité des systèmes.

Dès les années 90, les développeurs et les utilisateurs des systèmes SCADA ont considéré l'aspect sécuritaire comme une priorité de très haut niveau. Plusieurs organisations nationales et internationales ont été créées, des lois et des textes juridiques ont été établis et des normes, des standards et des stratégies nationaux, régionaux et internationaux ont été conçus. L'analyse d'une quinzaine de ces standards et de plusieurs stratégies de cyber-sécurité nous a montré que les aspects sécuritaires des réseaux de contrôles industriels et des réseaux IT ont été confondus dans la majorité des cas alors qu'ils présentent des différences majeures en termes d'exigences, de qualité de services, d'objectifs et d'architectures sécuritaires et de compositions matérielles et logicielles. De ce fait, aucune stratégie n'a pris en compte les spécificités des segments réseaux de coopération utilisés par les réseaux de contrôles industriels.

Suite à cette analyse, nous avons développé le nouveau protocole de transport de données hautement sécurisé le ST-101 basé sur l'IEC 60870-5-101, le protocole SCADA ouvert le plus répandu en Afrique du nord, en Europe, en Chine et dans plusieurs régions du monde. Ce dernier protocole a été choisi pour ses qualités notamment : sa simplicité, sa richesse, sa robustesse et ses mises à jour périodiques de sa couche applicative. Le ST-101 se base sur l'introduction d'une couche Sécurité entre la couche physique et la couche liaison de l'architecture EPA de l'IEC 60870-5-101. Cette couche assure une implémentation pratique de la confidentialité parfaite et la confidentialité idéale forte de Shannon.

Nous avons conçu cette couche à partir d'un nouveau crypto-système qui satisfait les contraintes temporelles du système SCADA. Elle incorpore deux sous-couches : La sous-couche Egalisateur de la distribution et la sous-couche Cryptogramme. La sous-couche Egalisateur de la distribution implémente le principe de la confidentialité idéale forte par la modification des caractéristiques probabilistes des messages envoyés étalés sur 261 octets de longueur. Cette sous-couche ajoute un bruit pseudo-aléatoire étudié aux trames IEC-101 de longueurs variables, certaines trames de longueur fixe et d'autres caractères de contrôle du signal pour créer les trames T-101 longues modifiées de 261 octets.

La sous-couche Cryptogramme est une implémentation pratique du principe de la confidentialité parfaite et du cryptage par masque jetable. Elle alterne entre trois blocs de cryptage par XOR et trois autres blocs de cryptage par transposition. Chaque bloc de cryptage sélectionne la clé de cryptage à partir d'une table de clés en se basant sur la longueur originale de la trame IEC-101 à crypter, sur les valeurs des octets de synchronisation, et sur la valeur du checksum de la trame ST-101 créée.

La sous-couche Cryptogramme génère périodiquement de nouvelles tables de clés de codage par XOR et de codage par transposition en se basant sur la clé de l'utilisateur, le poids, le moment actuel et un moment prédéfini. Pour obtenir une forte pseudo-indépendance et un haut niveau de confidentialité des clés des tables, nous avons choisi d'utiliser le CS-PRNG Blum-Blum-Shub, le générateur de nombre pseudo-aléatoire hautement sécurisé et l'algorithme de hachage sécurisé Skein (rapide, simple, sécurisé, flexible et le fiable finaliste du concours SHA-3) pour générer les tables de clés. Cette sous-couche utilise, aussi, une fonction de modification du temps d'exécution pour augmenter, intentionnellement, la complexité et le temps de génération des clés. Cette implémentation assure la vérification de l'authenticité du système et le protège contre les attaques passives et les attaques par modification et par fabrication.

En plus des résultats pratiques qui montrent son respect aux contraintes temporelles, notre protocole ST-101 fournit un haut niveau de sécurité par l'usage de plusieurs mécanismes tels que : (1) Ajouter suffisamment de bruit pseudo-aléatoire pour garantir une distribution uniforme des messages transmis,(2) coder certaines trames de longueur fixe et de caractères de contrôle du signal pour camoufler les trames de longueurs variables porteuses d'importantes données SCADA, (3) marquer le moment de fabrication des messages envoyés pour se protéger contre la réutilisation des trames,(4) produire des clés de codage pseudo-indépendantes garanties par l'usage de forts CS-PRNG et fonctions de hachage, (5) utiliser une fonction de modification du temps d'exécution, (6) sélectionner des clés de cryptage dépendamment d'un ensemble de paramètres indépendants, (7) changer souvent les clés de cryptage, et finalement (8) ignorer les messages qui présentent des problèmes lors du décryptage.

Finalement, nous avons présenté dans ce travail une plate-forme réelle pour le développement d'autres protocoles de transport de données. Il suffit d'adapter les sous-couches égalisation de la distribution et Cryptogramme aux caractéristiques statistiques et probabiliste des trames échangées.

Des travaux futurs sont envisagés pour optimiser le processus de génération du bruit et la fonction de modification du temps d'exécution, étudier d'autres CS-PRNG et fonctions de hachage pour générer des clés pseudo-indépendantes, proposer une version compacte pour les équipements de ressources limitées, introduire, adapter et tester notre approche avec d'autres protocoles SCADA ouverts tels que IEC 60870-5-104, DNP3, MODBUS et IEC-61850

Bibliographie

Alcaraz C, Zeadally S (2015) Critical infrastructure protection: Requirements and challenges for the 21st century. International Journal of Critical Infrastructure Protection 8:53-66

Aumasson J-P, Çalık Ç, Meier W, Özen O, Phan RC-W, Varıcı K (2009) Improved cryptanalysis of Skein Advances in Cryptology–ASIACRYPT 2009. Springer, p 542-559

Bailey D, Wright E (2003) Practical SCADA for industry. Newnes

Barker E, Barker W, Burr W, Polk W, Smid M, Gallagher PD (2012) NIST Special Publication 800-57 Recommendation for Key Management–Part 1: General.

Barker WC, Barker E (2016) NIST Special Publication 800-67 revision 1: Recommendation for the triple data encryption algorithm (TDEA) block cipher. PDF) Retrieved:10-09

Battistelli C, McKeever P, Gross S, Ponci F, Monti A (2018) Implementing energy service automation using cloud technologies and public communications networks Sustainable cloud and energy services. Springer, p 49-84

Bellare M, Kohno T, Lucks S, et al. (2009) Provable security support for the Skein hash family. Online–http://skein-hash info

Blum L, Blum M, Shub M (1986) A simple unpredictable pseudo-random number generator. SIAM Journal on computing 15(2):364-383

Branstad D, Gait J, Katzke S (1977) Report of the workshop on cryptography in support of computer security,

Brooks S, Nadeau E, Garcia M, Lefkovitz N, Lightman S (2015) Privacy Risk Management for Federal Information Systems. NIST Draft IR 8062:2015-05

Bui DM, Lien K-Y, Chen S-L, Cheng X-Y, Lin M-S (2016) Standards Commonly Used for Microgrids—A Research Project to Develop an Industry Microgrid Standard in Taiwan. Electric Power Components and Systems 44(19):2143-2160

Byres E, Eng P, Fellow I (2012) Using ANSI/ISA-99 standards to improve control system security. White paper, Tofino Security

Chang S-j, Perlner R, Burr WE, et al. (2012) Third-round report of the SHA-3 cryptographic hash algorithm competition. NIST Interagency Report 7896

Cherifi T, Hamami L (2018) A practical implementation of unconditional security for the IEC 60780-5-101 SCADA protocol

International Journal of Critical Infrastructure Protection 20C:68-84 doi:10.1016/j.ijcip.2017.12.001

Chien E, O'Murchu L, Falliere N W32. Duqu: The Precursor to the Next Stuxnet. In: LEET, 2012.

Choi K-s, Scott T, LeClair DP (2016) Ransomware against police: diagnosis of risk factors via application of cyber-routine activities theory. International Journal of Forensic Science & Pathology

Clarke GR, Reynders D, Wright E (2004) Practical modern SCADA protocols: DNP3, 60870.5 and related systems. Newnes

Cleveland F (2012) IEC TC57 WG15: IEC 62351 security standards for the power system information infrastructure. White Paper

Cyberattacks GE (2011) Night dragon. McAfee Foundstone Professional Services and McAfee Labs

Dacey RF (2004) Critical Infrastructure Protection: Challenges and efforts to secure control systems (Testimony Before the Subcommittee on Technology Information Policy, Intergovernmental Relations and the Census, House Committee on Government Reform).

Danger J-L, Guilley S, Hoogvorst P (2009) High speed true random number generator based on open loop structures in FPGAs. Microelectronics journal 40(11):1650-1656
Daniela T Communication security in SCADA pipeline monitoring systems. In: Roedunet International Conference (RoEduNet), 2011 10th, 2011. IEEE, p 1-5
Denning D (2007) A view of cyberterrorism 5 years later. Internet security: Hacking, counterhacking, and society:123-139
Denning DE (2000) Cyberterrorism: The logic bomb versus the truck bomb. Global Dialogue 2(4):29
Diffie W (1982) Cryptographic technology: fifteen year forecast. ACM SIGACT News 14(4):38-57
Drias Z, Serhrouchni A, Vogel O Analysis of cyber security for industrial control systems. In: Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC), 2015 International Conference on, 2015. IEEE, p 1-8
Drutarovsky M, Galajda P A robust chaos-based true random number generator embedded in reconfigurable switched-capacitor hardware. In: Radioelektronika, 2007 17th International Conference, 2007. IEEE, p 1-6
Ďuďák J, Gašpar G, Šedivý Š (2016) Securing communication layer of uBUS protocol Advanced Mechatronics Solutions. Springer, p 19-24
Energy UDo (2005) 21 Steps to Improve Cyber Security of SCADA Networks. White Paper
Erdbrink T, Nakashima E (28/09/2010) Iran struggling to contain 'foreign-made' computer worm The Washington Post.
Erez N, Wool A (2015) Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems. International Journal of Critical Infrastructure Protection 10:59-70
Faisal M, Ibrahim M (2012) Stuxnet, duqu and beyond. International Journal of Science and Engineering Investigations 1(2):75-78
Falliere N, Murchu LO, Chien E (2011) W32. stuxnet dossier. White paper, Symantec Corp, Security Response 5(6):29
Farwell JP, Rohozinski R (2011) Stuxnet and the future of cyber war. Survival 53(1):23-40
Fayi SYA (2018) What Petya/NotPetya Ransomware Is and What Its Remidiations Are Information Technology-New Generations. Springer, p 93-100
Ferguson N (1999) Impossible differentials in Twofish. Counterpane Systems October 19
Ferguson N, Lucks S, Schneier B, et al. (2010) The skein hash function family (version 1.3). Submitted to NIST SHA-3 Competition Round 3
Ferguson N, Schneier B (2003) Practical cryptography, vol 23. Wiley New York
Finco G, Lee K, Miller G, Tebbe J, Wells R (2007) Cyber Security Procurement Language for Control Systems Version 1.6. INL Critical Infrastructure Protection/Resilience Center, Idaho Falls, USA
Fischer V, Drutarovský M True random number generator embedded in reconfigurable hardware. In: International Workshop on Cryptographic Hardware and Embedded Systems, 2002. Springer, p 415-430
Gao J, Liu J, Rajan B, et al. (2014) SCADA communication and security issues. Security and Communication Networks 7(1):175-194
Genge B, Graur F, Haller P (2015) Experimental assessment of network design approaches for protecting industrial control systems. International Journal of Critical Infrastructure Protection 11:24-38
Gennaro R An improved pseudo-random generator based on discrete log. In: Annual International Cryptology Conference, 2000. Springer, p 469-481

Goldenberg N, Wool A (2013) Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. International Journal of Critical Infrastructure Protection 6(2):63-75
Gorman S (2009) Electricity grid in US penetrated by spies. The Wall Street Journal 8
Hao X, Zhou F, Chen X Analysis on security standards for industrial control system and enlightenment on relevant Chinese standards. In: Industrial Electronics and Applications (ICIEA), 2016 IEEE 11th Conference on, 2016. IEEE, p 1967-1971
Hartmann K, Giles K UAV exploitation: A new domain for cyber power. In: Cyber Conflict (CyCon), 2016 8th International Conference on, 2016. IEEE, p 205-221
Hartmann K, Steup C The vulnerability of UAVs to cyber attacks-An approach to the risk assessment. In: Cyber Conflict (CyCon), 2013 5th International Conference on, 2013. IEEE, p 1-23
Heaven D (2018) Gossip gives warnings for cyberattacks. Elsevier
Heginbotham E, Nixon M, Morgan FE, et al. (2015) The US-China military scorecard: Forces, geography, and the evolving balance of power, 1996–2017. Rand Corporation
Hellman M, Merkle R, Schroeppel R, Washington L, Diffie W, Pohlig S (1976) Results of an initial attempt to cryptanalyze the NBS Data Encryption Standard,
Huang Y-L, Cárdenas AA, Amin S, Lin Z-S, Tsai H-Y, Sastry S (2009) Understanding the physical and economic consequences of attacks on control systems. International Journal of Critical Infrastructure Protection 2(3):73-83
Huitsing P, Chandia R, Papa M, Shenoi S (2008) Attack taxonomies for the Modbus protocols. International Journal of Critical Infrastructure Protection 1:37-44
IEC_TC57 (1995) Transmission protocol - Section 101: Companion standard for basic telecontrol tasks. IEEE
Igure VM, Laughter SA, Williams RD (2006) Security issues in SCADA networks. Computers & Security 25(7):498-506
Jang SW (2017) Comparative Analysis of AES, Blowfish, Twofish and Threefish Encryption Algorithms. ANALYSIS OF APPLIED MATHEMATICS:5
Jerman-Blažič B, Schneider WS, Klobučar T (2001) Advanced security technologies in networking, vol 178. IOS press
Jun B, Kocher P (1999) The Intel random number generator. Cryptography Research Inc white paper
Karn P, Simpson WA, Metzger P (1995) The ESP triple DES transform.
Keizer G (2011) Sloppy'Chinese hackers scored data-theft coup with'Night Dragon'. Computer World.
Khovratovich D, Nikolić I Rotational cryptanalysis of ARX. In: International Workshop on Fast Software Encryption, 2010. Springer, p 333-346
Khovratovich D, Nikolić I, Rechberger C (2014) Rotational rebound attacks on reduced Skein. Journal of Cryptology 27(3):452-479
Knapp ED, Langill JT (2014) Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems. Syngress
Knowles W, Prince D, Hutchison D, Disso JFP, Jones K (2015) A survey of cyber security management in industrial control systems. International Journal of Critical Infrastructure Protection 9:52-80
Knudsen LR (1994) Block ciphers: analysis, design and applications. DAIMI Report Series 23(485)
Kong JH, Ang L-M, Seng KP (2015) A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. Journal of Network and Computer Applications 49:15-50

Kraft MB, Marks E (2016) US Government Counterterrorism: A guide to who does what. CRC Press
Kriaa S (2016) Joint safety and security modeling for risk assessment in cyber physical systems. Université Paris-Saclay
Kriaa S, Bouissou M, Piètre-Cambacédès L Modeling the Stuxnet attack with BDMP: Towards more formal risk assessments. In: Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on, 2012. IEEE, p 1-8
Kushner D (2013) The real story of stuxnet. ieee Spectrum 50(3):48-53
Langner R (2011) Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security & Privacy 9(3):49-51
Lee W-K, Cheong H-S, Phan RC-W, Goi B-M (2016) Fast implementation of block ciphers and PRNGs in Maxwell GPU architecture. Cluster Computing 19(1):335-347
Leyden J (2011) "Chinese Cyberspies" Target Energy Giants. The Register, February 10
Liu C-C, Stefanov A, Hong J, Panciatici P (2012) Intruders in the grid. IEEE Power and Energy magazine 10(1):58-66
Lu Z, Lu X, Wang W, Wang C Review and evaluation of security threats on the communication networks in the smart grid. In: MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010, 2010. IEEE, p 1830-1835
Luiijf E, Besseling K, De Graaf P (2013) Nineteen national cyber security strategies. International Journal of Critical Infrastructures 6 9(1-2):3-31
Luiijf H, Besseling K, Spoelstra M, De Graaf P Ten national cyber security strategies: A comparison. In: International Workshop on Critical Information Infrastructures Security, 2011. Springer, p 1-17
Markoff J (October 26, 2009) Cyberwar: Old Trick Threatens the Newest Weapons The New York Times.
Mathur M, Kesarwani A Comparison between Des, 3des, Rc2, Rc6, Blowfish And Aes. In: Proceedings of National Conference on New Horizons in IT-NCNHIT, 2013. vol 3. p 143-148
McDonald JD (2012) Electric power substations engineering. CRC press
Melton R, Fletcher T, Earley M System protection profile-industrial control systems. In: Version 10, National Institute of Standards and Technology, 2004.
Menezes AJ, Van Oorschot PC, Vanstone SA (1996) Handbook of applied cryptography. CRC press
Meunier P (2010) Algebre avec applications à l'algorithmique et à la cryptographie. Ellipses, France
MIIT PR (2016) Guide to industrial control systems information security protection. P.R. MIIT
Miller B, Rowe D A survey SCADA of and critical infrastructure incidents. In: Proceedings of the 1st Annual conference on Research in information technology, 2012. ACM, p 51-56
ModbusIDA (2004) Modbus messaging on TCP/IP implementation guide V 1.0b v1 0b, October 2006. vol 4. Modbus Organization, Inc.
Morris R (1978) The data encryption standard--Retrospective and prospects. IEEE Communications Society Magazine 16(6):11-14
Morris R, Sloane N, Wyner AD (1977) Assessment of the National Bureau of Standards proposed federal data encryption standard. Cryptologia 1(3):281-291
Mustard S (2005) Security of distributed control systems: The concern increases. Computing & Control Engineering Journal 16(6):19-25
Nazir S, Patel S, Patel D (2017) Assessing and augmenting SCADA cyber security: A survey of techniques. Computers & Security 70:436-454
Nicholson A, Webber S, Dyer S, Patel T, Janicke H (2012) SCADA security in the light of Cyber-Warfare. Computers & Security 31(4):418-436

Perlroth N, Scott M, Frenkel S (2017) Cyberattack Hits Ukraine Then Spreads Internationally. The New York Times
Petitcolas FA (2011) Kerckhoffs' principle Encyclopedia of cryptography and security. Springer, p 675-675
Piggin R Emerging good practice for cyber security of Industrial Control Systems and SCADA. In: System Safety, incorporating the Cyber Security Conference 2012, 7th IET International Conference on, 2012. IET, p 1-6
Piggin R Development of industrial cyber security standards: IEC 62443 for SCADA and Industrial Control System security. In: Control and Automation 2013: Uniting Problems and Solutions, IET Conference on, 2013. IET, p 1-6
Provos N, Mazieres D A Future-Adaptable Password Scheme. In: USENIX Annual Technical Conference, FREENIX Track, 1999. p 81-91
Reddy TC, Seshadri R (2013) New Design of Crypto-Based Pseudo random number generator (CBPRNG) using BLOW FISH cipher. International Journal on Computer Science and Engineering 5(6):561
Robinson M, Jones K, Janicke H (2015) Cyber warfare: Issues and challenges. Computers & Security 49:70-94
Salmon D, Zeller M, Guzman A, Mynam V, Donolo M Mitigating the aurora vulnerability with existing technology. In: proceedings of the 36th Annual Western Protective Relay Conference, 2009.
Saravanan K, Anusuya E, Kumar R (2018) Real-time water quality monitoring using Internet of Things in SCADA. Environmental monitoring and assessment 190(9):556
Schneier B Description of a new variable-length key, 64-bit block cipher (Blowfish). In: Fast Software Encryption, 1993. Springer, p 191-204
Schneier B (1994) Applied Cryptography--Protocols, Algorithms, and.
Schneier B (2005) Twofish Cryptanalysis Rumors. Schneier on Security blog
Schneier B, Kelsey J, Whiting D, Wagner D, Hall C, Ferguson N (1998) Twofish: A 128-bit block cipher. NIST AES Proposal 15
Schneier B, Kelsey J, Whiting D, Wagner D, Hall C, Ferguson N (1999) The Twofish encryption algorithm: a 128-bit block cipher. John Wiley & Sons, Inc.
Shahzad A, Lee M, Lee C, et al. (2015) The protocol design and New approach for SCADA security enhancement during sensors broadcasting system. Multimedia Tools and Applications:1-28
Shakarian P, Shakarian J, Ruef A (2013) Introduction to cyber-warfare: A multidisciplinary approach. Newnes
Shannon CE (1949) Communication theory of secrecy systems*. Bell system technical journal 28(4):656-715
Sidorenko A, Schoenmakers B Concrete security of the blum-blum-shub pseudorandom generator. In: IMA International Conference on Cryptography and Coding, 2005. Springer, p 355-375
Skoko V, Atlagic B, Isakov N Comparative realization of IEC 60870-5 industrial protocol standards. In: Telecommunications Forum Telfor (TELFOR), 2014 22nd, 2014. IEEE, p 987-990
Smaiah S, Khellaf A, Cherifi T The implementation of SCADA open protocol IEC60870-5-101 on ARDUINO UNO board. In: Electrical Engineering (ICEE), 2015 4th International Conference on, 2015a. IEEE, p 1-6
Smaiah S, Khellaf A, Cherifi T The implementation of SCADA open protocol PUR 2.4. In: New Technologies of Information and Communication (NTIC), 2015 First International Conference on, 2015b. IEEE, p 1-6

Sommestad T, Ericsson GN, Nordlander J SCADA system cyber security—A comparison of standards. In: Power and Energy Society General Meeting, 2010 IEEE, 2010. IEEE, p 1-8
Standard N-F (2001) Announcing the advanced encryption standard (AES). Federal Information Processing Standards Publication 197:1-51
Stouffer K, Falco J, Kent K (2006) Guide to Supervisory Control and Data Acquisition (SCADA) and industrial control systems security–initial public draft. National Institute of Standards and Technology, Gaithersburg, Maryland
Stouffer K, Falco J, Scarfone K (2011) Guide to industrial control systems (ICS) security. NIST Special Publication 800(82):16-16
Stouffer K, Falco J, Scarfone K (2014) Guide to Industrial Control Systems (ICS) Security. NIST Special Publication 800:82
Sunar B, Martin WJ, Stinson DR (2007) A provably secure true random number generator with built-in tolerance to active attacks. IEEE Transactions on computers 56(1)
Tama J (2016) Maximizing the Value of Quadrennial Strategic Planning. Center for The Business of Government
Tkacik TE A hardware random number generator. In: International Workshop on Cryptographic hardware and embedded systems, 2002. Springer, p 450-453
Trappey AJ, Trappey CV, Govindarajan UH, Chuang AC, Sun JJ (2017) A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. Advanced Engineering Informatics 33:208-229
Tripathi S, Gupta B, Almomani A, Mishra A, Veluru S (2013) Hadoop based defense solution to handle distributed denial of service (ddos) attacks. Journal of Information Security 4(03):150
Tsang R (2010) Cyberthreats, vulnerabilities and attacks on SCADA networks. University of California, Berkeley, Working Paper, http://gspp berkeley edu/iths/Tsang_SCADA% 20Attacks pdf (as of Dec 28, 2011)
Wei D, Lu Y, Jafari M, Skare P, Rohde K An integrated security system of protecting smart grid against cyber attacks. In: Innovative Smart Grid Technologies (ISGT), 2010, 2010. IEEE, p 1-7
Weiss J (2016) Aurora generator test Handbook of SCADA/Control Systems Security, Second Edition. CRC Press, p 142-149
Zeller M Myth or reality—Does the Aurora vulnerability pose a risk to my generator? In: Protective Relay Engineers, 2011 64th Annual Conference for, 2011. IEEE, p 130-136
Zhou X, Xu Z, Wang L, Chen K What should we do? A structured review of SCADA system cyber security standards. In: Control, Decision and Information Technologies (CoDIT), 2017 4th International Conference on, 2017. IEEE, p 0605-0614
Zhu B, Joseph A, Sastry S A taxonomy of cyber attacks on SCADA systems. In: Internet of things (iThings/CPSCom), 2011 international conference on and 4th international conference on cyber, physical and social computing, 2011. IEEE, p 380-388