Etude expérimentale de la sécurité de WirelessHART

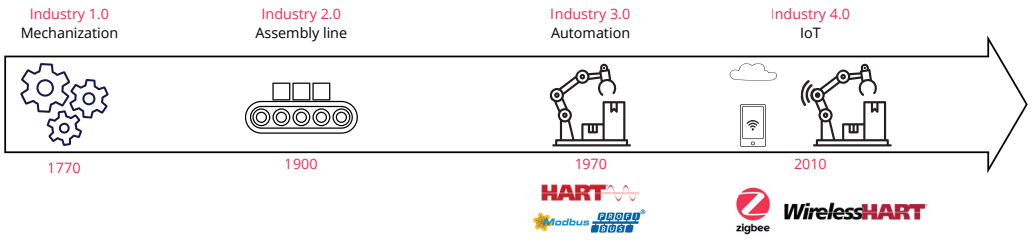
Romain Cayre[1, 2] **Kais Sellami**[1]
Elies Tali[2] Pierre Ayoub[2] Vincent Nicomette[1, 2] Guillaume Auriol[1, 2]

[1]Univ. Toulouse, INSA, France
[2]LAAS-CNRS, Toulouse, France

Juin 2026

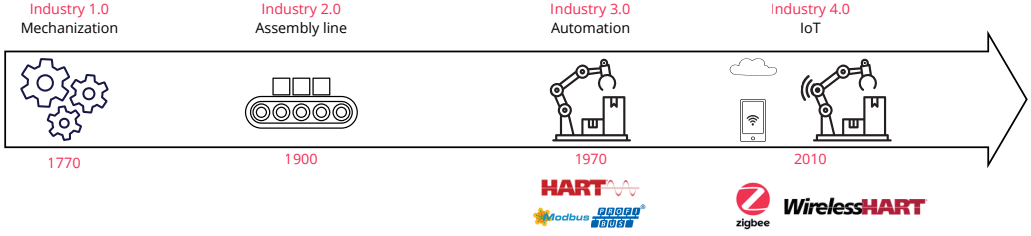
Contexte industriel

- **Evolution de l'industrie**
  - De l'automatisation à l'industrie 4.0
  - Numérisation et interconnexion
  - Emergence de l'internet industriel des objets (IIoT)
- **Protocoles industriels**
  - Protocoles filaires: Modbus, PROFIBUS, HART
  - Passage à l'échelle limité



Contexte industriel

- **Evolution de l'industrie**
  - De l'automatisation à l'industrie 4.0
  - Numérisation et interconnexion
  - Emergence de l'internet industriel des objets (IIoT)
- **Protocoles industriels**
  - Protocoles filaires: Modbus, PROFIBUS, HART
  - Passage à l'échelle limité
- **Adoption de technologies sans fil**
  - WirelessHART, ISA100.11a, Zigbee
  - Déploiement flexible, coût réduit
- **Challenges**
  - Sécurité des systèmes (cyber attaques, intégrité des données)
  - Coexistence avec les systèmes existants



Durant la présentation

Focus sur WirelessHART

- Sniffer
  - Les fondamentaux de WirelessHART
  - Les communications de WirelessHART
  - Implémentation
- Injecteur : attaques
  - Déauthentification massive
  - Désynchronisation temporelle

Cas d'usage : WirelessHART

Pharmaceutique

Cas d'usage : WirelessHART

Pharmaceutique

Petrochimie

Cas d'usage : WirelessHART

Pharmaceutique

Petrochimie

Energie

Cas d'usage : WirelessHART

Pharmaceutique

Petrochimie

Energie

Acier

Problématique

- Recherche limitée
  - Moins étudié que d'autres protocoles
  - Peu d'études expérimentales

Problématique

- Recherche limitée
  - Moins étudié que d'autres protocoles
  - Peu d'études expérimentales
- Absence d'implémentation open-source
  - Aucune pile protocolaire accessible
  - Difficile à reproduire et expérimenter

Problématique

- Recherche limitée
  - Moins étudié que d'autres protocoles
  - Peu d'études expérimentales
- Absence d'implémentation open-source
  - Aucune pile protocolaire accessible
  - Difficile à reproduire et expérimenter
- Outils limités
  - Absence d'outils d'écoute/injection à bas coût
  - Recherches basées sur la simulation

Problématique

- Recherche limitée
  - Moins étudié que d'autres protocoles
  - Peu d'études expérimentales
- Absence d'implémentation open-source
  - Aucune pile protocolaire accessible
  - Difficile à reproduire et expérimenter
- Outils limités
  - Absence d'outils d'écoute/injection à bas coût
  - Recherches basées sur la simulation

Objectifs

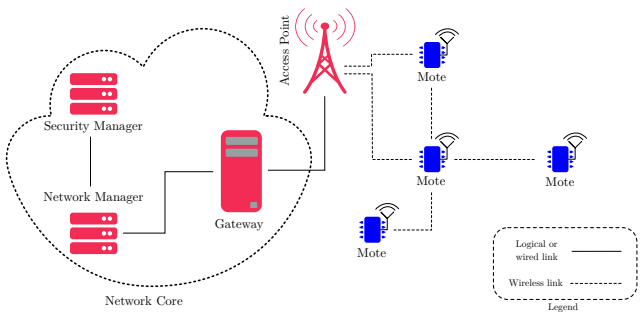
- Concevoir un sniffer open-source à bas coût
- Permettre l'injection de trafic contrôlé
- Evaluer des attaques pratiques sur le protocole

Les fondamentaux de WirelessHART

- Bande ISM radio 2.4 GHz
  → Sans licence
- Basé sur la couche physique
  IEEE 802.15.4
  → Faible consommation

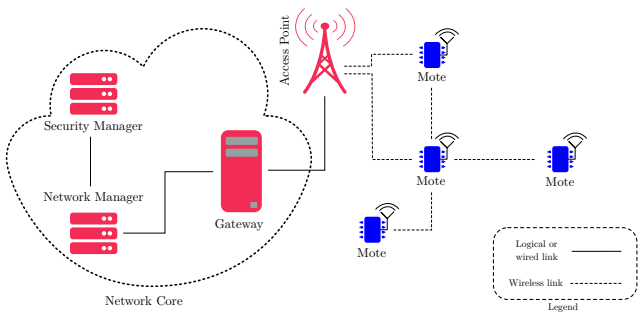
Les fondamentaux de WirelessHART

- Bande ISM radio 2.4 GHz
  → Sans licence
- Basé sur la couche physique IEEE 802.15.4
  → Faible consommation
- Réseau maillé
  → Tolérance aux pannes



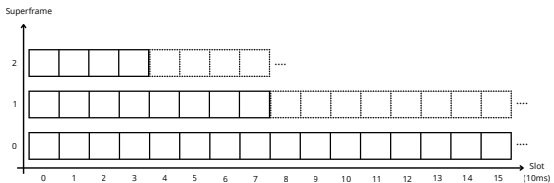
Les fondamentaux de WirelessHART

- Bande ISM radio 2.4 GHz
  → Sans licence
- Basé sur la couche physique IEEE 802.15.4
  → Faible consommation
- Réseau maillé
  → Tolérance aux pannes
- Time Division Multiple Access (TDMA)
  → Déterministe
- Frequency Hopping Spread Spectrum (FHSS)
  → Résistant face aux interférences



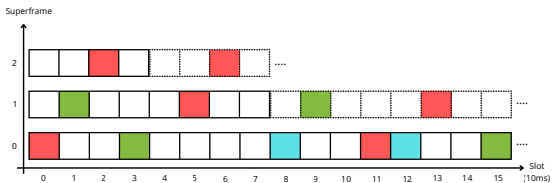
Les communications de WirelessHART

- **Division temporelle:** recoupage en slots de 10ms regroupés en *superframes*.



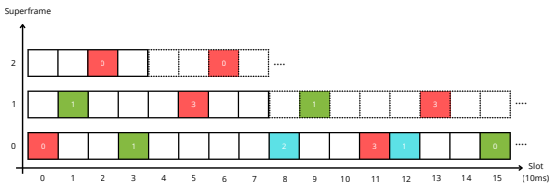
Les communications de WirelessHART

- **Division temporelle:** recoupage en slots de 10ms regroupés en *superframes*.
- **Liens:** communications planifiées



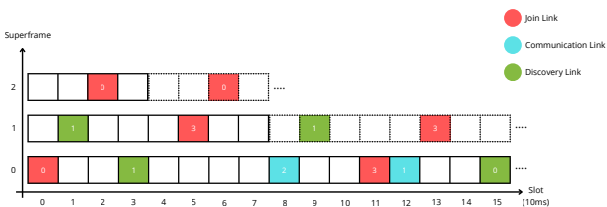
Les communications de WirelessHART

- **Division temporelle:** recoupage en slots de 10ms regroupés en *superframes*.
- **Liens:** communications planifiées
  - Décrites par: superframe ID, index de Slot, **offset**, type et des options.



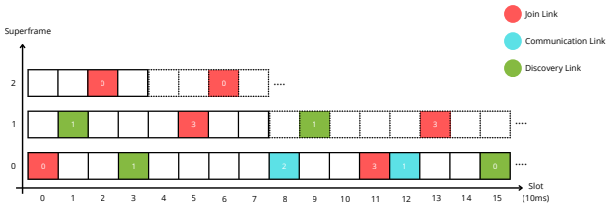
Les communications de WirelessHART

- **Division temporelle:** recoupage en slots de 10ms regroupés en *superframes*.
- **Liens:** communications planifiées
  - Décrites par: superframe ID, index de Slot, **offset**, type et des options.
  - Types: Association, Broadcast, Découverte and Communication normale.



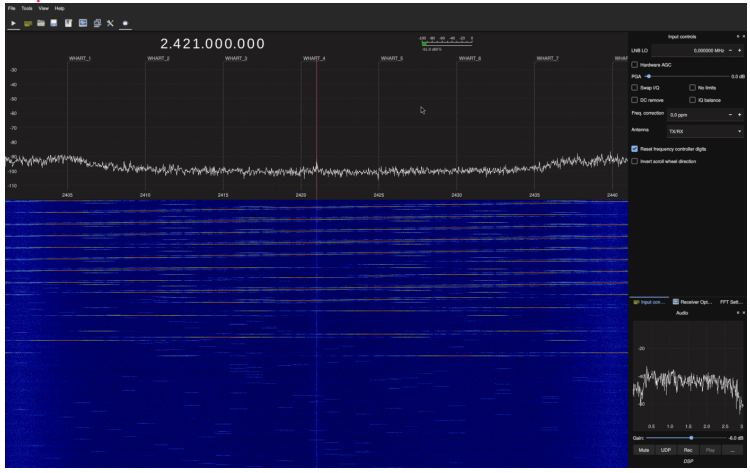
Les communications de WirelessHART

- **Division temporelle:** recoupage en slots de 10ms regroupés en *superframes*.
- **Liens:** communications planifiées
  - Décrites par: superframe ID, index de Slot, **offset**, type et des options.
  - Types: Association, Broadcast, Découverte and Communication normale.
- **Gestion:** Seul le *Network Manager* est responsable de la configuration du réseau, l'ordonnancement des communication et la gestion de la table de routage.



Les communications de WirelessHART

Analyseur de spectre



Les communications de WirelessHART

- **Fréquences utilisées:** Utilisation des canaux 11 à 25 (IEEE 802.15.4) déclaré par une *channel map* (2 octets).
- **Algorithme de saut de fréquence:**

$$\text{index} = (\text{ASN} + \text{offset}) \bmod N_{\text{used}} \tag{1}$$

ASN = Absolute Slot Number
N_{used} = *nombredecanauxutilisés*

Les communications de WirelessHART

- **Fréquences utilisées:** Utilisation des canaux 11 à 25 (IEEE 802.15.4) déclaré par une *channel map* (2 octets).
- **Algorithme de saut de fréquence:**

$$\text{index} = (\text{ASN} + \text{offset}) \bmod N_{\text{used}} \tag{1}$$

$$\text{channel} = \text{activeChannelArray}[\text{index}] \tag{2}$$

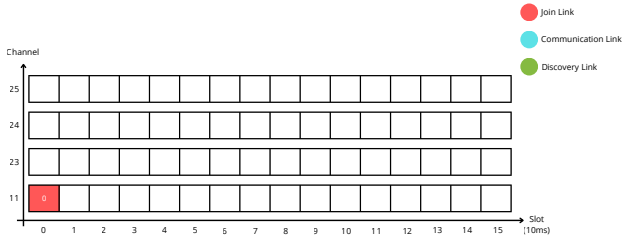
ASN = Absolute Slot Number
N_{used} = *nombredecanauxutilisés*

Les communications de WirelessHART

- **Fréquences utilisées:** Utilisation des canaux 11 à 25 (IEEE 802.15.4) déclaré par une *channel map* (2 octets).
- **Algorithme de saut de fréquence:**

$$\text{index} = (\text{ASN} + \text{offset}) \bmod N_{\text{used}} \tag{1}$$

$$\text{channel} = \text{activeChannelArray}[\text{index}] \tag{2}$$



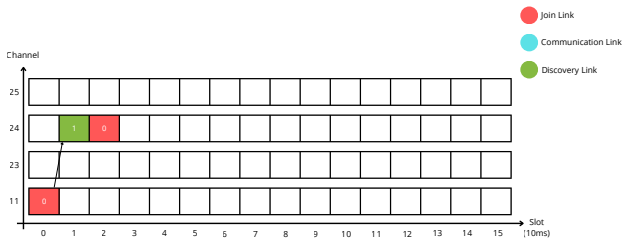
ASN = Absolute Slot Number
N_{used} = *nombredecanauxutilisés*

Les communications de WirelessHART

- **Fréquences utilisées:** Utilisation des canaux 11 à 25 (IEEE 802.15.4) déclaré par une *channel map* (2 octets).
- **Algorithme de saut de fréquence:**

$$\text{index} = (\text{ASN} + \text{offset}) \bmod N_{\text{used}} \tag{1}$$

$$\text{channel} = \text{activeChannelArray}[\text{index}] \tag{2}$$



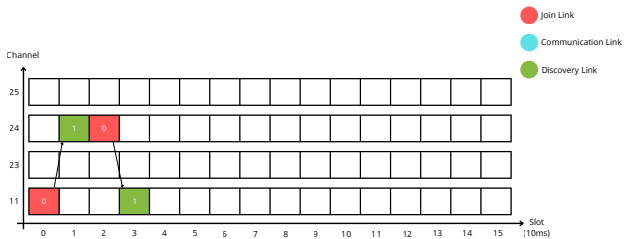
ASN = Absolute Slot Number
N_{used} = *nombredecanauxutilisés*

Les communications de WirelessHART

- **Fréquences utilisées:** Utilisation des canaux 11 à 25 (IEEE 802.15.4) déclaré par une *channel map* (2 octets).
- **Algorithme de saut de fréquence:**

$$\text{index} = (\text{ASN} + \text{offset}) \bmod N_{\text{used}} \tag{1}$$

$$\text{channel} = \text{activeChannelArray}[\text{index}] \tag{2}$$



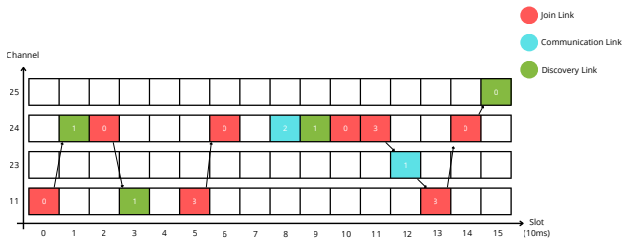
ASN = Absolute Slot Number
N_{used} = nombredecanauxutilisés

Les communications de WirelessHART

- **Fréquences utilisées:** Utilisation des canaux 11 à 25 (IEEE 802.15.4) déclaré par une *channel map* (2 octets).
- **Algorithme de saut de fréquence:**

$$\text{index} = (\text{ASN} + \text{offset}) \bmod N_{\text{used}} \tag{1}$$

$$\text{channel} = \text{activeChannelArray}[\text{index}] \tag{2}$$



ASN = Absolute Slot Number
N_{used} = *nombredecanauxutilisés*

WHAD – Wireless HAcking Devices

Framework open-source de cybersécurité sans fil

- Supporte BLE, ZigBee, LoRaWAN...
- Interface radio unifiée hôte/matériel

Trois composants

- `whad-client` – hôte
- `ButteRFly` – un firmware du dongle
- `whad-protocol` – protocole de communication

Sniffer WirelessHART

Slots de 10 ms / fenêtre de réception de 2 ms
⇒ traitement côté hôte non réalisable

ButteRFly (firmware)

- Calcul du canal actif
- Changement de canal radio
- Timer hardware pour la synchronisation des slots
- Ajustement temporel à partir du champ Time adjustment contenu dans les ACKs
- Capture de paquet & transfert via `whad-protocol`

whad-client (Python)

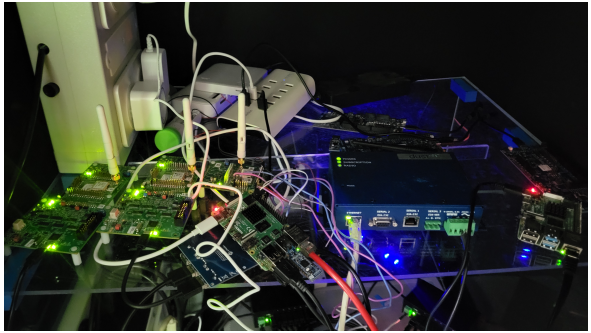
- Chiffrement & déchiffrement des paquets
- Injection de paquets
- Inférence des liens non observés (algorithme d'exploration)
- Dissection WirelessHART via **Scapy**
- Logging & export

Setup expérimental : Dust Network

- **Matériel**
  - Dust Networks (Analog Devices)
- **Composition du réseau**
  - Une gateway: Network Manager + Security Manager + point d'accès (AP)
  - Plusieurs motes

Evaluation du Sniffer

Expérience	Durée	Environnement	Nombre de motes	Trafic	Sniffer	**Total**
A	14 h	Laboratoire	Gateway + 2 motes	Normal	$Snif_4$	14601
					$Snif_{FH}$	14593
					Taux de perte	**0.05%**
B	45 mins	Domestique	Gateway + 5 motes	Ping continu	$Snif_4$	25500
					$Snif_{FH}$	22288
					Taux de perte	**12.6%**



Du Sniffing aux attaques

- Prochaine étape: **Attaques**
- Réaliser les attaques nécessite la compréhension de:
  - la communication dans un réseau WirelessHART
  - la sécurité appliquée

→ Communication: les phases et la securité

Phases de communication

- **Phase d'annonce**
  - Le point d'accès annonce les *liens d'association*
  - Le mote écoute et se synchronise

Phases de communication

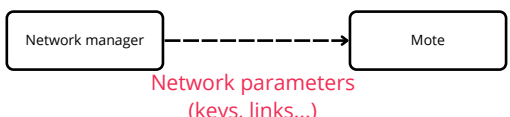
- **Phase d'annonce**
  - Le point d'accès annonce les *liens d'association*
  - Le mote écoute et se synchronise
- **Phase d'association**
  - Le mote envoit une *requête d'association* au Network Manager

Phases de communication

- **Phase d'annonce**
  - Le point d'accès annonce les *liens d'association*
  - Le mote écoute et se synchronise
- **Phase d'association**
  - Le mote envoit une *requête d'association* au Network Manager
  - Le Network Manager authentifie le mote

Phases de communication

- **Phase d'annonce**
  - Le point d'accès annonce les *liens d'association*
  - Le mote écoute et se synchronise
- **Phase d'association**
  - Le mote envoit une *requête d'association* au Network Manager
  - Le Network Manager authentifie le mote
  - Allocation de liens de communication et des paramètres du réseau



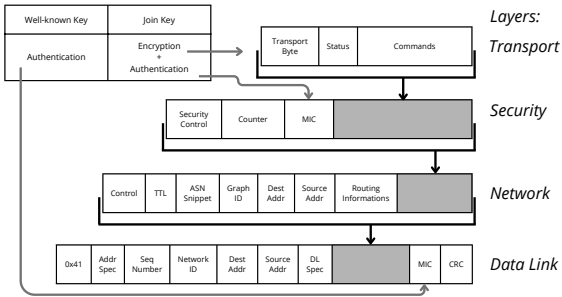
Phases de communication

- **Phase d'annonce**
  - Le point d'accès annonce les *liens d'association*
  - Le mote écoute et se synchronise
- **Phase d'association**
  - Le mote envoit une *requête d'association* au Network Manager
  - Le Network Manager authentifie le mote
  - Allocation de liens de communication et des paramètres du réseau
- **Phase de communication sécurisée**
  - Communications ordonnancées
  - Données chiffrées et authentifiées

La sécurité de WirelessHART

Clés cryptographiques

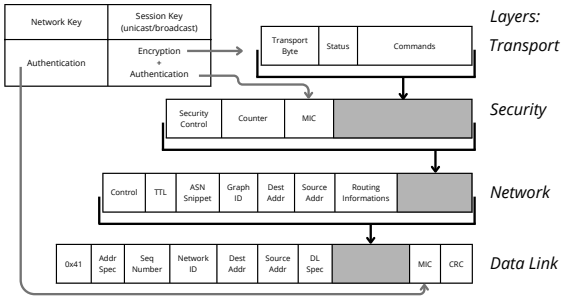
- **Well-known key:**
  - utilisée pour le calcul du MIC lors des annonces et de l'association à la couche de liaison de données (DLL)
- **Join key:**
  - prépartagée
  - sécurise l'association (handshake): chiffrement + authentification



La sécurité de WirelessHART

Clés cryptographiques

- **Network key:**
  - partagée entre tout le réseau (DLL)
  - utilisée pour l'authentification
- **Session Keys:**
  - **Unicast:** confidentialité entre les motes
  - **Broadcast:** transmission à tout le réseau



La sécurité de WirelessHART

Basée sur AES-128 CCM*

- **Confidentialité:** chiffrement des données avec les clés de sessions

La sécurité de WirelessHART

Basée sur AES-128 CCM*

- **Confidentialité:** chiffrement des données avec les clés de sessions
- **Integrité:** utilisation du MIC (Message Integrity Code)
  - **NWK-MIC:** integrité des données chiffrées à la couche réseau
  - **DL-MIC:** integrité du PDU à la couche de liaison de données (DLL)

La sécurité de WirelessHART

Basée sur AES-128 CCM*

- **Confidentialité:** chiffrement des données avec les clés de sessions
- **Integrité:** utilisation du MIC (Message Integrity Code)
  - **NWK-MIC:** integrité des données chiffrées à la couche réseau
  - **DL-MIC:** integrité du PDU à la couche de liaison de données (DLL)
- **Anti-rejeu:** utilisation du nonce derivé de l'ASN

Modèle de menace

Hypothèses

- Accès actif & passif au radio
- Compromission de la clé K_{join} (trash-can attack, clé par défaut/faible/prédictible)
- Ecoute passive d'une association pour récupérer les autres clés

Modèle de menace

Objectifs de l'attaquant

- **Confidentialité:** interception des données des capteurs
- **Authenticité:** injection de paquets malveillants en usurpant l'identité d'un mote légitime
- **Disponibilité:** suspendre un ou plusieurs motes

Extraction de clé K_{join}

Trash can attack

Attaques OTA

Déauthentification massive

Désynchronisation temporelle

Déauthentification massive

Principe

- Exploite la commande Suspend (opcode `0x3CC`)
- Ordonne à un mote de se suspendre entre deux ASNs
- Broadcast $\Rightarrow$ suspension simultanée de plusieurs motes

Nécessite

- K_{network}, $K_{\text{session\_broadcast}}$
- La connaissance des liens de communication

Déauthentification massive

Impact

- Déni de service partiel ou total du réseau sans fil
- Perturbation des boucles de contrôle & des processus industriels
- Peut nécessiter une **intervention manuelle**

Evaluation

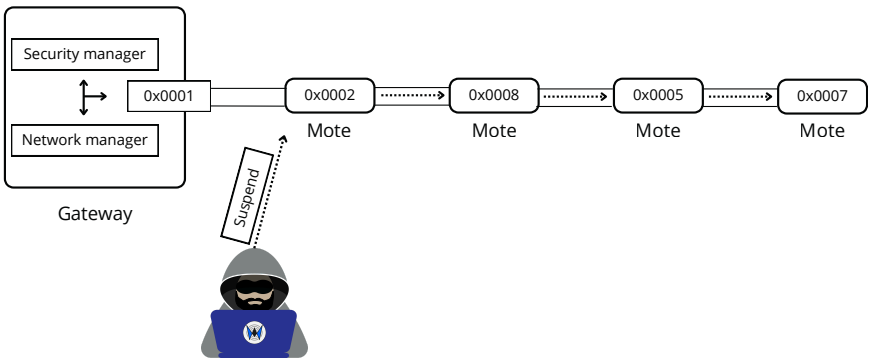
- Theorisé en 2015[a]
- Jamais évalué faute de manque d'outils offensifs

[a] Duijsens, Master thesis

Scénarios de déauthentification massive

Déni de service

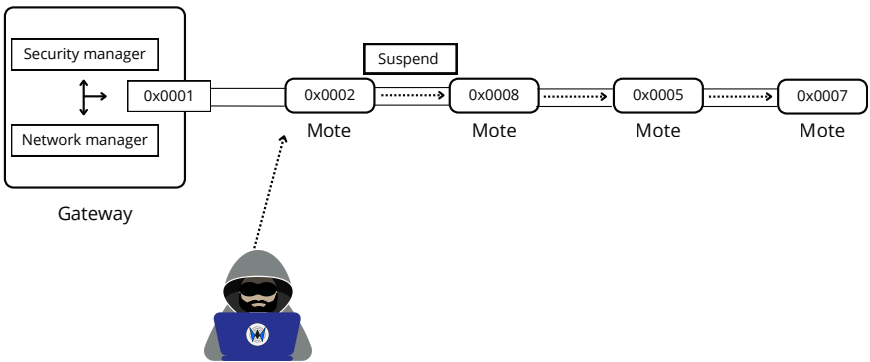
- Transmission d'une commande Suspend pour une durée de 1M de slots $\simeq$ 170 mins



Scénarios de déauthentification massive

Déni de service

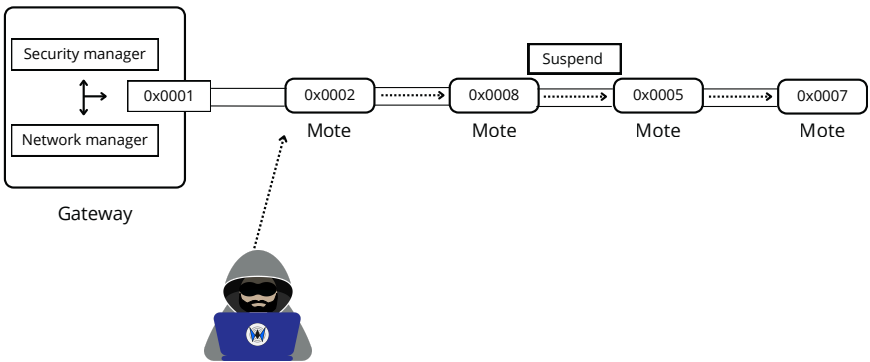
- Transmission d'une commande Suspend pour une durée de 1M de slots $\triangleq$ 170 mins
- Requête routée



Scénarios de déauthentification massive

Déni de service

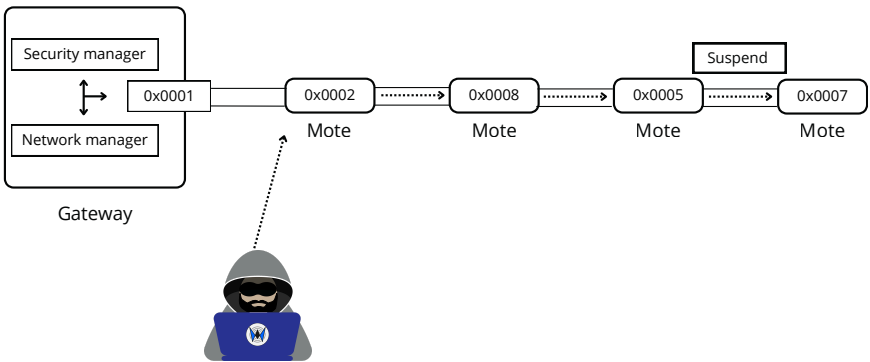
- Transmission d'une commande Suspend pour une durée de 1M de slots $\triangleq$ 170 mins
- Requête routée



Scénarios de déauthentification massive

Déni de service

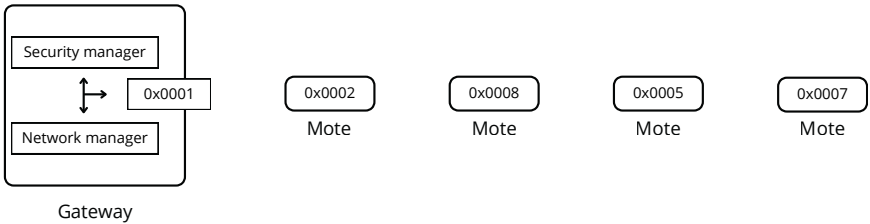
- Transmission d'une commande Suspend pour une durée de 1M de slots $\triangleq$ 170 mins
- Requête routée



Scénarios de déauthentification massive

Déni de service

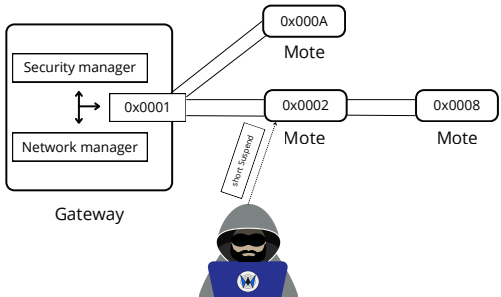
- Transmission d'une commande Suspend pour une durée de 1M de slots $\triangleq$ 170 mins
- Requête routée
- Réseau déconnecté



Scénarios de déauthentification massive

Réassociation forcée

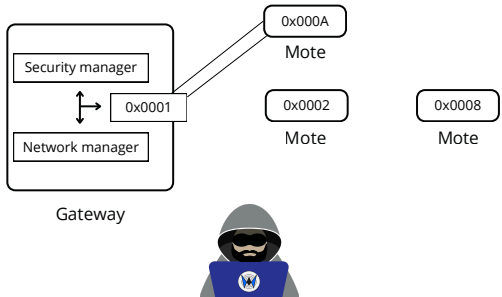
- Transmission de commande Suspend pour une courte durée



Scénarios de déauthentification massive

Réassociation forcée

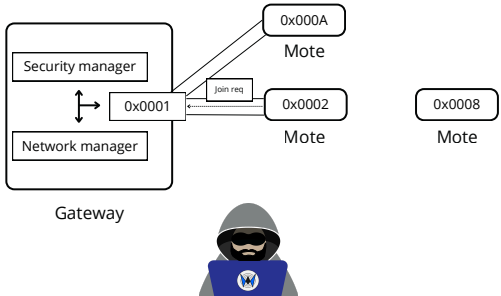
- Transmission de commande Suspend pour une courte durée



Scénarios de déauthentification massive

Réassociation forcée

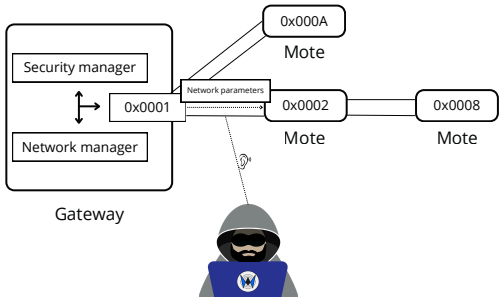
- Transmission de commande Suspend pour une courte durée
- Reconnexion automatique des motes une fois la durée du suspend finie: dépendence d'implémentation



Scénarios de déauthentification massive

Réassociation forcée

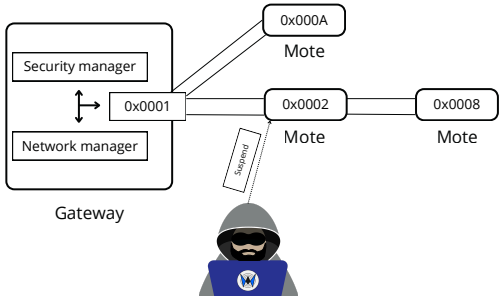
- Transmission de commande Suspend pour une courte durée
- Reconnexion automatique des motes une fois la durée du suspend finie: dépendence d'implémentation
- Récupération des clés manquantes du mote visé par l'attaquant suite à l'association



Scénarios de déauthentification massive

Réassociation forcée

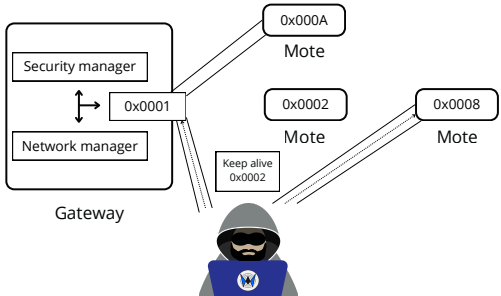
- Transmission de commande Suspend pour une courte durée
- Reconnexion automatique des motes une fois la durée du suspend finie: dépendence d'implémentation
- Récupération des clés manquantes du mote visé par l'attaquant suite à l'association
- Usurpation d'identité par envoi d'une commande Suspend avec un long délai



Scénarios de déauthentification massive

Réassociation forcée

- Transmission de commande Suspend pour une courte durée
- Reconnexion automatique des motes une fois la durée du suspend finie: dépendence d'implémentation
- Récupération des clés manquantes du mote visé par l'attaquant suite à l'association
- Usurpation d'identité par envoi d'une commande Suspend avec un long délai



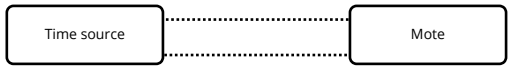
Démo - déauthentification massive

Attaque déauthentification massive

Désynchronisation temporelle

Voisins

Deux motes qui communiquent sur la couche de liaison de données (DLL)
Pour chaque paire de voisins, il existe une source de temps



Neighbors

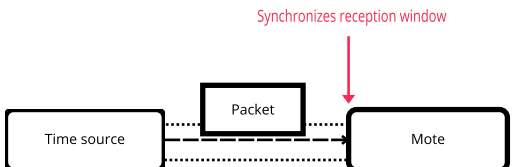
Désynchronisation temporelle

Voisins

Deux motes qui communiquent sur la couche de liaison de données (DLL)
Pour chaque paire de voisins, il existe une source de temps

Synchronisation

- Le temps de réception de la part de la source de temps



Neighbors

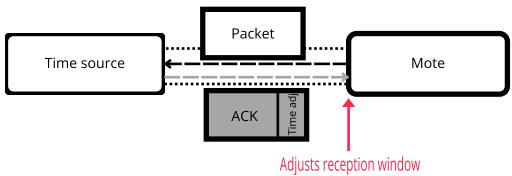
Désynchronisation temporelle

Voisins

Deux motes qui communiquent sur la couche de liaison de données (DLL)
Pour chaque paire de voisins, il existe une source de temps

Synchronisation

- Le temps de réception de la part de la source de temps
- Le champs **Time adjustment** contenu dans les ACKs



Neighbors

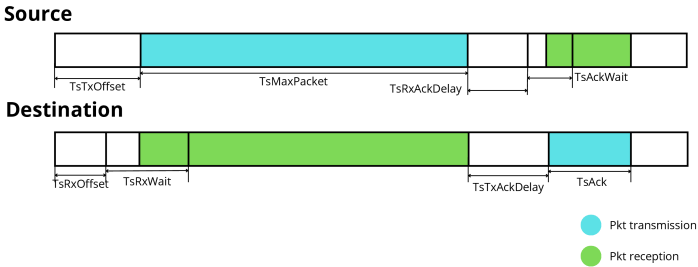
Désynchronisation temporelle

Principe

Usurper l'identité de la source de temps et envoyer du trafic différé afin de désynchroniser la fenêtre de réception (TsRxWait)

Impact

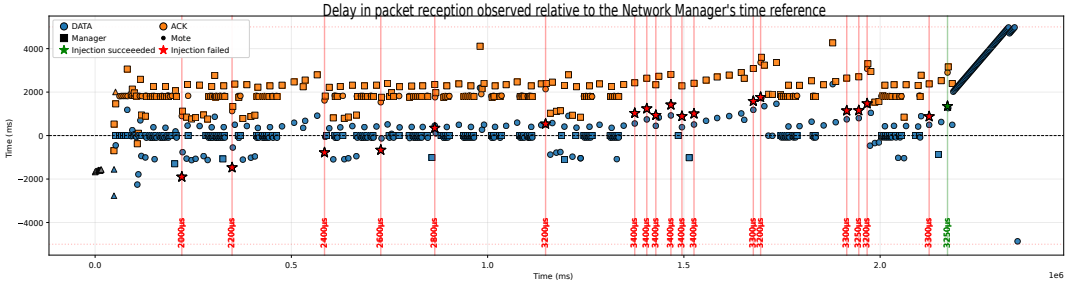
- Slot désynchronisé
- Saut de fréquence retardé
- Déni de service



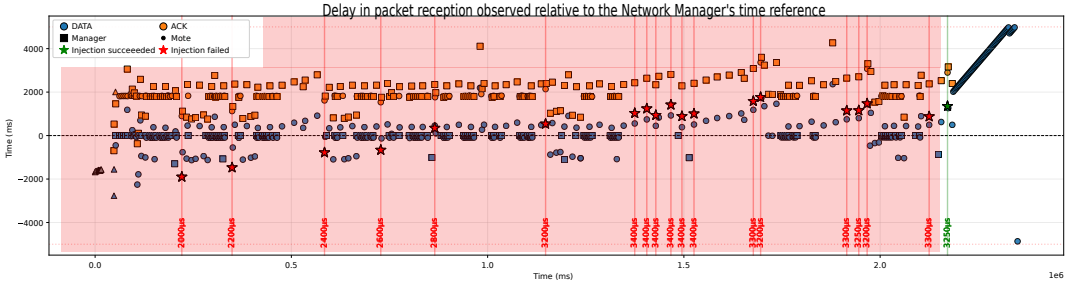
Démo - désynchronisation temporelle

Attaque de désynchronisation temporelle

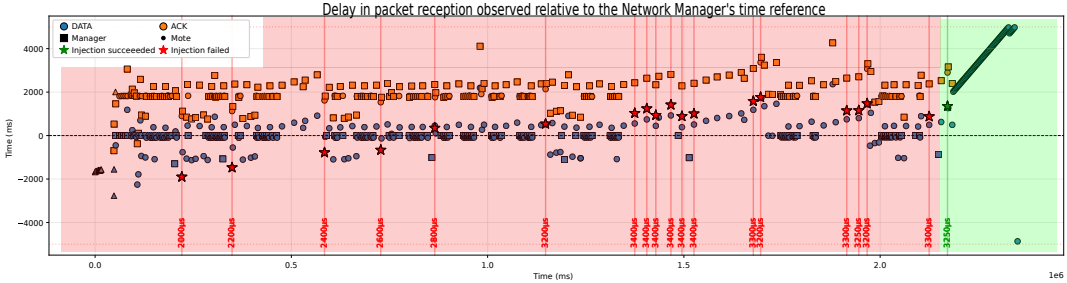
Evaluation de la désynchronisation temporelle



Evaluation de la désynchronisation temporelle



Evaluation de la désynchronisation temporelle



Contre-mesures

Clés K_{join} différentes

- Forcer l'utilisation d'une clé différente par mote
- Limiter la compromission à un unique mote

Contre-mesures

Clés K_{join} différentes

- Forcer l'utilisation d'une clé différente par mote
- Limiter la compromission à un unique mote

Interdire la commande Suspend en broadcast

- Nécessite peu de modifications des mécanismes du réseau
- Empêche l'interruption du réseau à partir d'un mote corrompu

Contre-mesures

Clés K_{join} différentes

- Forcer l'utilisation d'une clé différente par mote
- Limiter la compromission à un unique mote

Interdire la commande Suspend en broadcast

- Nécessite peu de modifications des mécanismes du réseau
- Empêche l'interruption du réseau à partir d'un mote corrompu

Détecter les ajustements temporels anormaux

- Détecter et prévenir les ajustements temporels suspects
- Rejeter les dérives d'horloge anormalement élevées

Conclusion & Perspectives

Conclusion

- Les travaux de recherche sur WirelessHART restent limités et théoriques
- Conception d'un outil d'analyse open-source accessible et économique
- Les risques d'attaques sont présents malgré la sécurité appliquée par le protocole

Conclusion & Perspectives

Conclusion

- Les travaux de recherche sur WirelessHART restent limités et théoriques
- Conception d'un outil d'analyse open-source accessible et économique
- Les risques d'attaques sont présents malgré la sécurité appliquée par le protocole

Perspectives

- Continuer l'étude de WirelessHART et tester d'autres attaques sur ce protocole
  - Proposer une implémentation open-source de la pile du réseau WirelessHART
  - Analyser les mécanismes de la gateway de dust network : mise à jour OTA...
- Investiguer les interactions IT/OT
- Etudier d'autres protocoles industriels sans fil : ISA100.11a

Merci!

WHAD-client

https://github.com/whad-team/whad-client

ButteRFly

https://github.com/whad-team/butterfly

Bibliographie

- A security analysis for wireless sensor mesh networks in highly critical systems
- Security analysis of WirelessHART communication scheme
- Security issue of WirelessHART based SCADA systems
- One for all and all for WHAD : Wireless shenanigans made easy
- Software-defined radio based measurement platform for wireless networks
- Cracking the channel hopping sequences in IEEE 802.15.4e-based industrial TSCH networks
- Revealing smart selective jamming attacks in WirelessHART networks
- Security considerations for the WirelessHART protocol
- WirelessHART : Applying wireless technology in real-time industrial process control
- WirelessHART : A Security Analysis
- FieldComm Group : WirelessHART documentation
- Formal verification of the WirelessHART protocol - verifying old and finding new attacks
- 802.15.4e in a nutshell : Survey and performance evaluation
- Reverse engineering WirelessHART hardware
- Formal security evaluation and improvement of WirelessHART protocol in industrial wireless network
- A comparison of WirelessHART and ISA100.11a
- A survey on the application of WirelessHART for industrial process monitoring and control
- Securing WirelessHART : Monitoring, exploring and detecting new vulnerabilities