Transmission sécurisée de trames ECG via UART en utilisant ASCON sur FPGA

Auteurs : Rawane AL ZOHBI & Halima GHANNOUM

Date : Mars 2025

Objectif du projet

Ce projet a pour but de sécuriser la transmission des signaux ECG d'un patient en les chiffrant avant leur envoi. Les données sont reçues par un port de communication et traitées sur une carte FPGA à l'aide de l'algorithme de chiffrement ASCON.

L'objectif principal est de transmettre via **UART** des données sensibles biomédicales (signaux ECG) à travers une interface matérielle, de les **chiffrer avec ASCON** dans le FPGA, puis de les **déchiffrer côté PC** en Python pour vérification et traitement.

Avantages de cette approche :

- **Sécurisation** de la transmission des données sensibles (ECG, identifiants, etc.).
- **Validation d'une architecture embarquée complète** combinant **UART + FSM + chiffrement ASCON**.
- **Possibilité de traitements avancés post-déchiffrement** pour l'analyse médicale des signaux ECG (détection d'arythmies, anomalies de conduction cardiaque, etc.).

Ce projet illustre un cas d'usage **concret et moderne** de la cryptographie légère, en lien direct avec les enjeux de sécurité dans les **dispositifs médicaux connectés**.

Architecture du projet

L'architecture repose sur une série de modules spécialisés interconnectés :

Modules FPGA :

- **uart_core.sv** : interface UART RX/TX assurant la communication avec le PC.
- **fsm_uart.sv** : machine d'état principale qui réceptionne les données envoyées (clé, nonce, trames ECG, données associées), les stocke dans des registres et déclenche le chiffrement.
- **ascon.sv** : cœur de chiffrement, activé par la FSM UART, qui traite les blocs en suivant les étapes du protocole ASCON.
- **fsm.sv** : FSM interne à ASCON, gérant les phases du chiffrement (initialisation, association, chiffrement, finalisation).
- **compteur.sv** : indexeur de blocs utilisé par la FSM ASCON pour envoyer chaque trame dans l'ordre.
- **ascon_reg.sv** : buffer de sortie stockant les blocs chiffrés avant de les renvoyer au PC via UART.

Fonctionnement en trois étapes :

1. **Réception UART** : les données (clé, nonce, données associées et trames ECG) sont envoyées depuis le terminal PC, reçues et stockées dans le FPGA.
2. **Chiffrement ASCON** : une fois les données disponibles, le cœur ASCON est déclenché. Il chiffre les trames et génère un tag d'authentification.
3. **Transmission sécurisée** : les résultats (trames chiffrées + tag) sont renvoyés vers le PC pour vérification ou traitement logiciel (ex. en Python).

L'ensemble du système est cadencé à **50 MHz** grâce à un module de conversion d'horloge (clk_wiz), garantissant une synchronisation correcte avec UART.

L'utilisation de Ip **ILA Debug** était utile pour déboguer les ports non connectés.

Outils et technologies

- **Vivado** : conception, simulation et synthèse FPGA.
- **HTerm** : interface UART pour tester les commandes.
- **Python** : post-traitement (déchiffrement et analyse ECG).
- **ILA / Logic Analyzer** : vérification des signaux internes.
- **FPGA PYNQ-Z2** : plateforme matérielle utilisée pour implémenter le système.

Limitations rencontrées

Problèmes rencontrés	**Description**
Compréhension de l'algorithme ASCON	Il a fallu bien assimiler son fonctionnement pour l'implémenter correctement.
Gestion des cycles d'horloge	Il était crucial d'éviter les cycles d'horloge vides pour empêcher un chiffrement excessif et un tag incorrect.
Communication UART	Assurer une transmission fiable des données entre les composants.
Débogage du module **InterSpartan**	Vérifier et corriger les erreurs pour garantir le bon fonctionnement du système.

Améliorations possibles

- **Autonomie énergétique** : Intégration d'une batterie pour alimenter le système, avec stockage du code, permettant une utilisation plus durable.
- **Affichage des données** : Ajouter un écran pour visualiser les données du patient en temps réel ou intégrer une solution IoT pour envoyer ces données au médecin pour un suivi à distance et l'alerte.
- **Intelligence Artificielle** : Utiliser l'IA pour prévoir et détecter tout dysfonctionnement ou anomalie dans le système, ainsi que pour analyser les tendances des données du patient.