Instituto Tecnológico de Estudios Superiores de Monterrey
Escuela de Ingeniería y Ciencias
Ingeniería en Ciencia de Datos y Matemáticas
Uso de álgebras modernas para seguridad y criptografía

Implementación de criptografía de clave pública para protección de comunicaciones y almacenamiento de datos con IoT en entornos de monitoreo y consumo de energía.

A00831314 Paola Sofía Reyes Mancheno
A01197399 Diana Paola Cadena Nito
A01275180 Alexis Hernández Spinola
A01285041 María Fernanda Torres Alcubilla
A01705747 Enrique García Varela
A01730548 Javier Hernández Arellano

LiCore
A 16 de junio de 2023 en Monterrey, Nuevo León

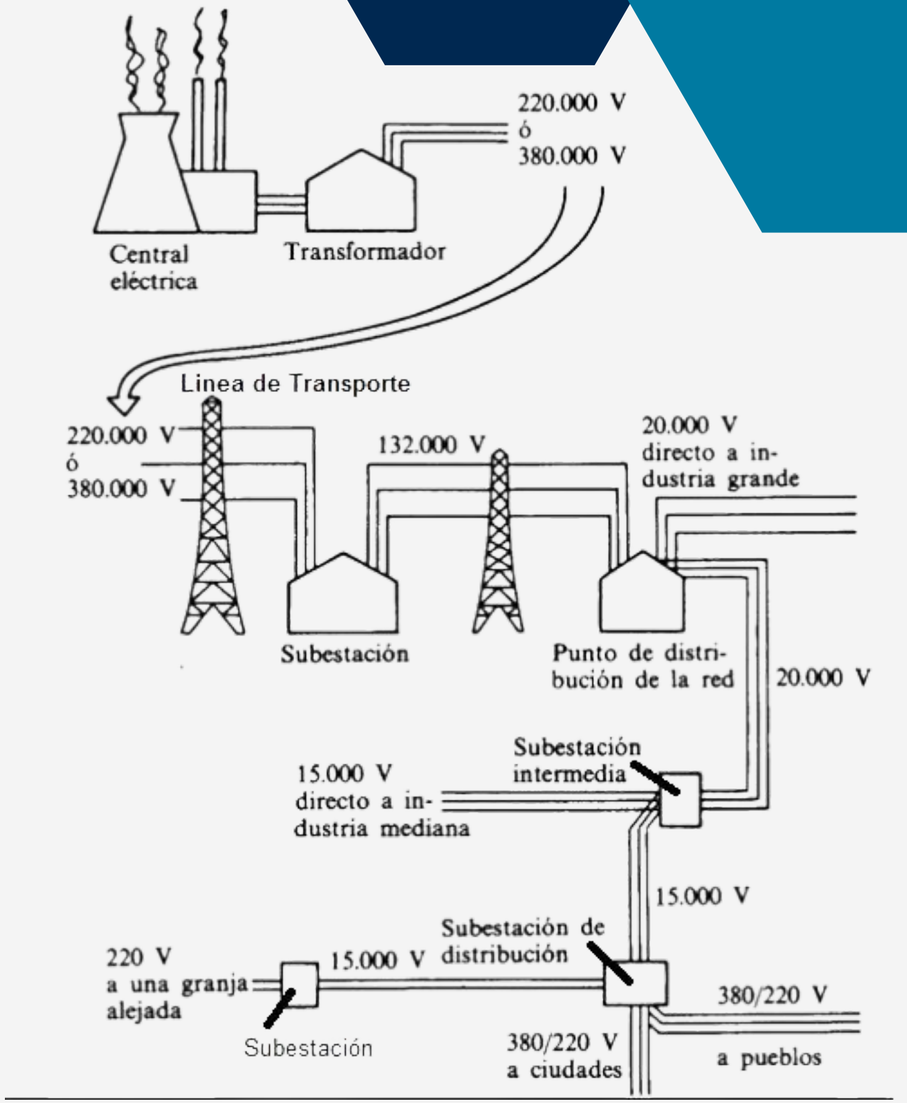
Contexto

Transición realista hacia tecnologías y combustibles más limpios dentro del contexto Mexicano con la **Generación Distribuida**

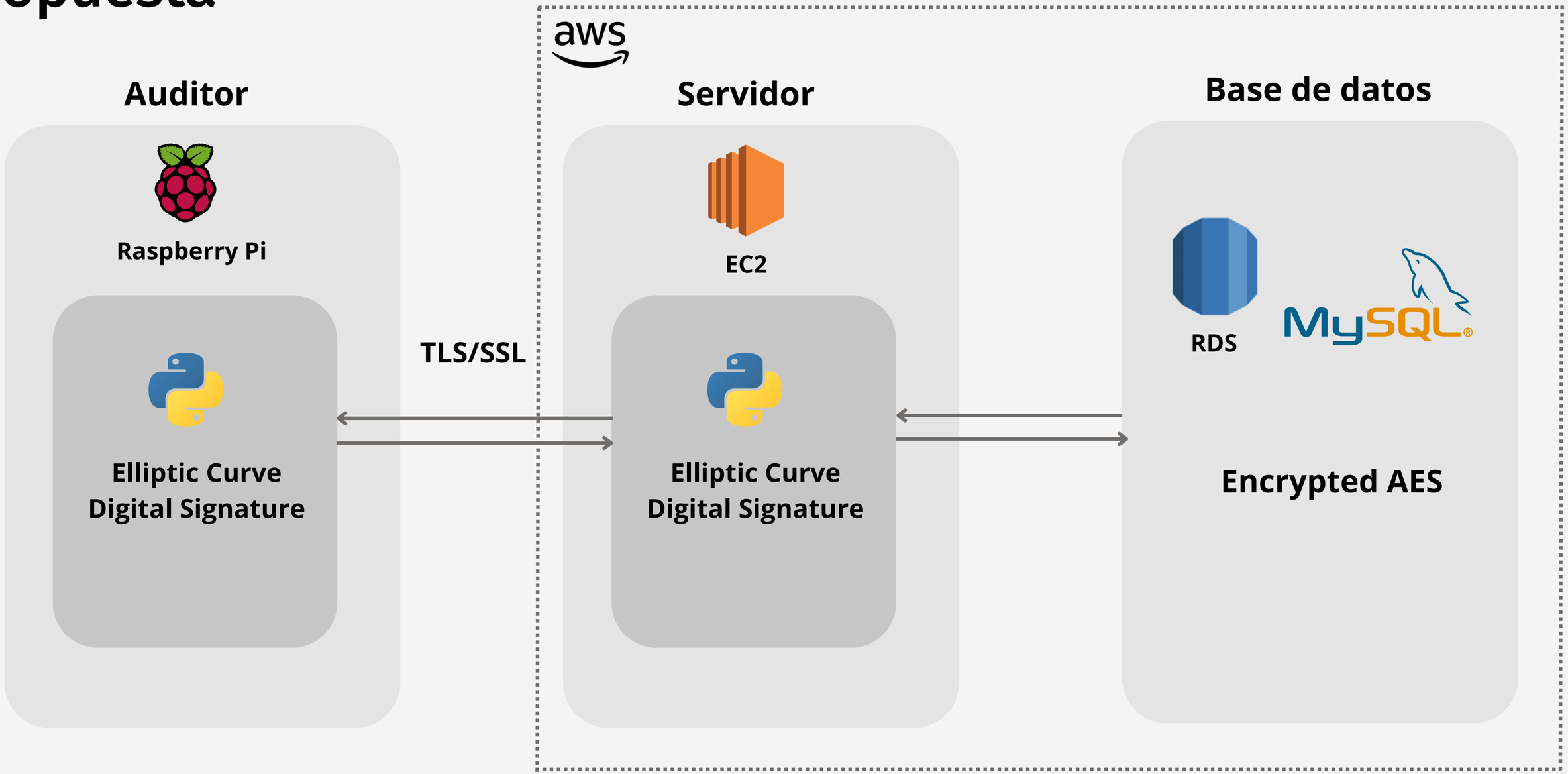
- Controlar oferta y demanda
- Herramientas inteligentes
- Recolección y monitoreo de datos en tiempo real
- Garantizar la integridad y confidencialidad de los datos con un sistema IoT

Objetivo

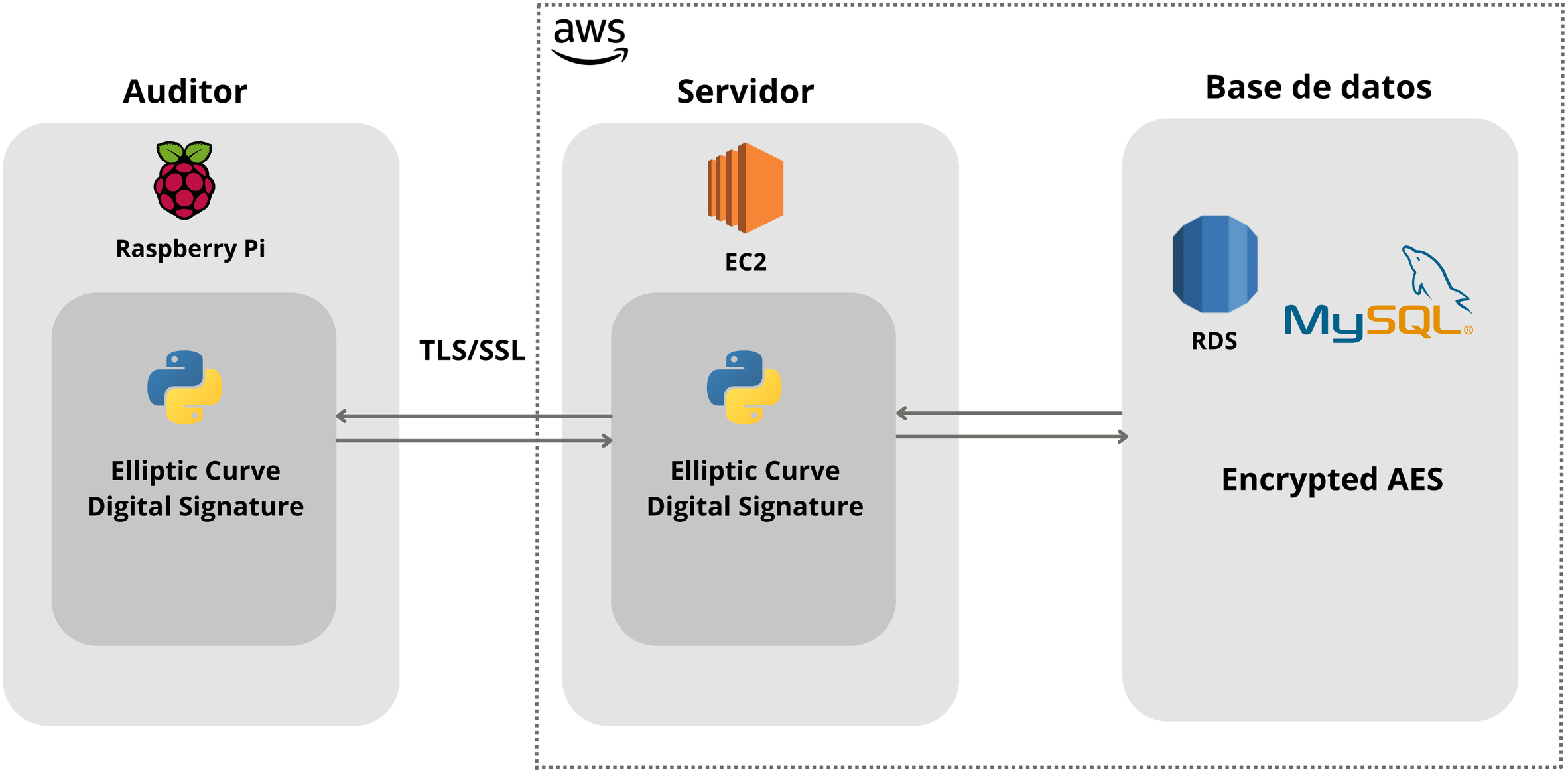
Crear un esquema de conexión que cumpla con los requisitos de conexiones seguras, donde los registros sean incorporados a una base de datos cada 15 minutos y también se encuentren seguros.



Propuesta



Link a video: https://youtu.be/4GtgOzOpeF8


aws
Auditor
Servidor
Base de datos
Raspberry Pi
EC2
RDS
MySQL
TLS/SSL
Elliptic Curve
Digital Signature
Elliptic Curve
Digital Signature
Encrypted AES

Justificación

Algoritmos de autenticación

Asimétricos y simétricos, populares y ligeros.

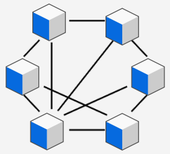
RSA, ECC, ElGamal, DSA, ECDSA ; AES, DES, 3DES, Blowfish, Serpent ; BlueJay, ECC, CLEFIA, SIMON, TRIVIUM, PHOTON.

Arquitecturas

Client-server

Edge-cloud

Peer to peer



Blockchain

IoT en la nube

Azure IoT Hub	Home Assistant	IBM Watson	AWS EC2
HTTP, MQTT, AMPQ, SQL Azure	*TCP, MQTT, InfluxDB*	*MQTT*	*TLS/SSL*

Bases de datos

SQL

NoSQL

Basado en el artículo IoT Privacy and Security: Challenges and Solutions por Texas A&M University

Proyección de impacto

	Inversión por año (USD)
Instancia tipo t3.small	$122.65
General Purpose SSD (gp3)	$2.4
Total	$125.05

Beneficios

- Autenticación
- Privacidad de la red y usuarios
- Acceso solo a personal autorizado
- Almacenamiento y respaldo de datos
- Pruebas de testeo

Implementación (Next steps)

Budget

Definir la cantidad de auditores iniciales y obtener presupuestos para EC2 y RDS.

Certificados

Comprar un dominio actual, asociarlo al servidor y obtener certificados a través de CA gratuita como Let's Encrypte. Evitando así, que estos sean autofirmados.

Interfaz web

Integración con app para la visualización de datos y análisis.

Auditor

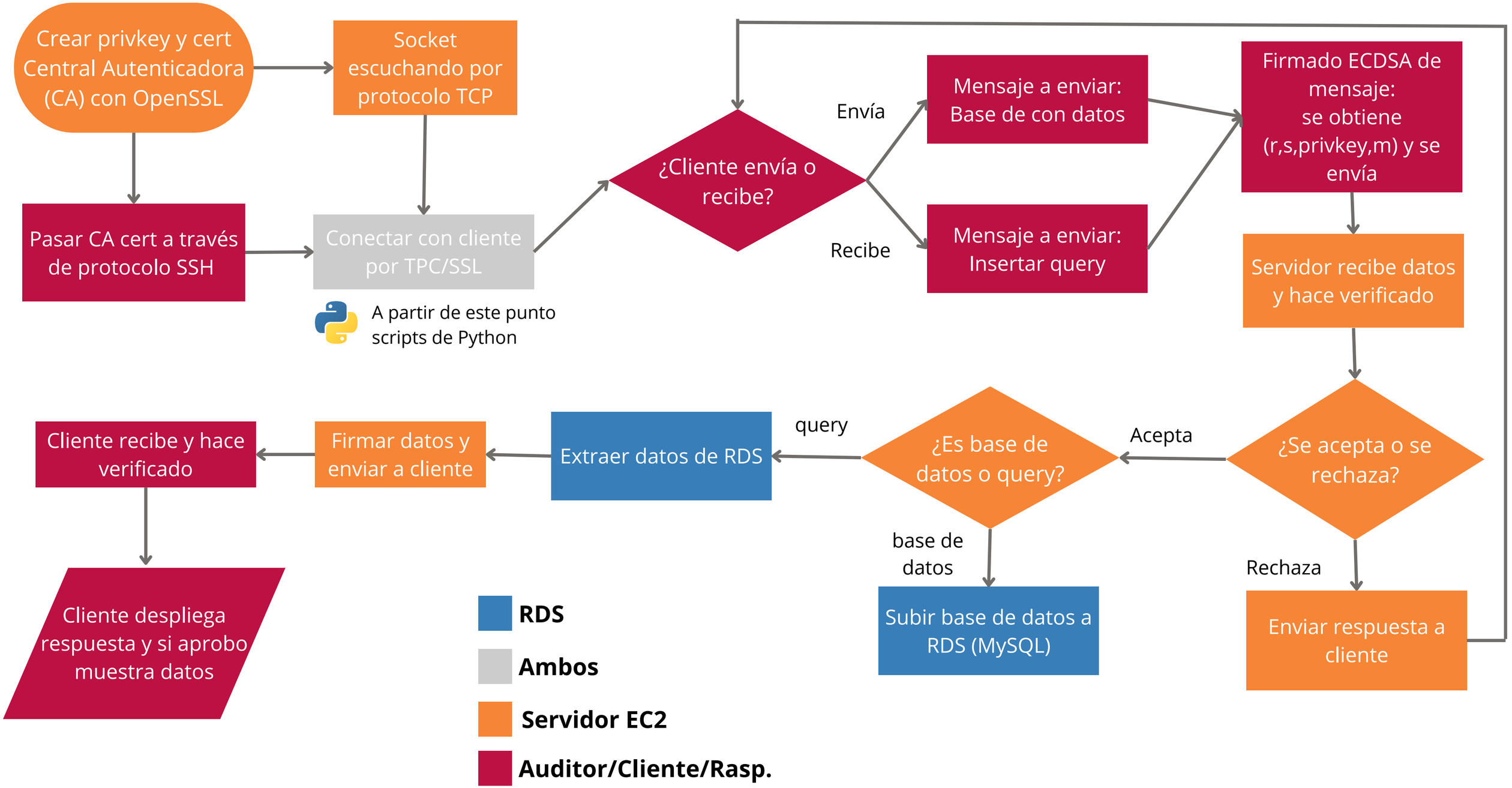
Copiar código actual en tarjetas físicas Raspberry y realizar conexión con sensores

Múltiples conexiones

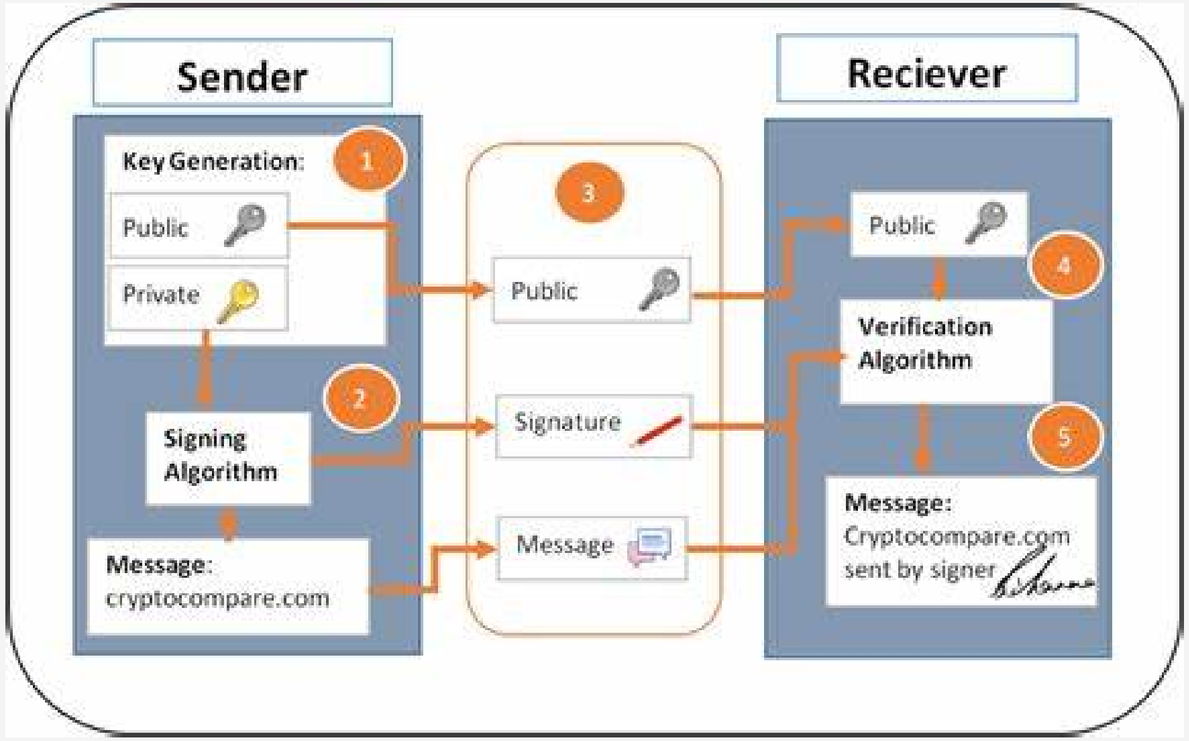
Habilitar más que solo a los usuarios a conectarse, el cliente no solo será el auditor en si.

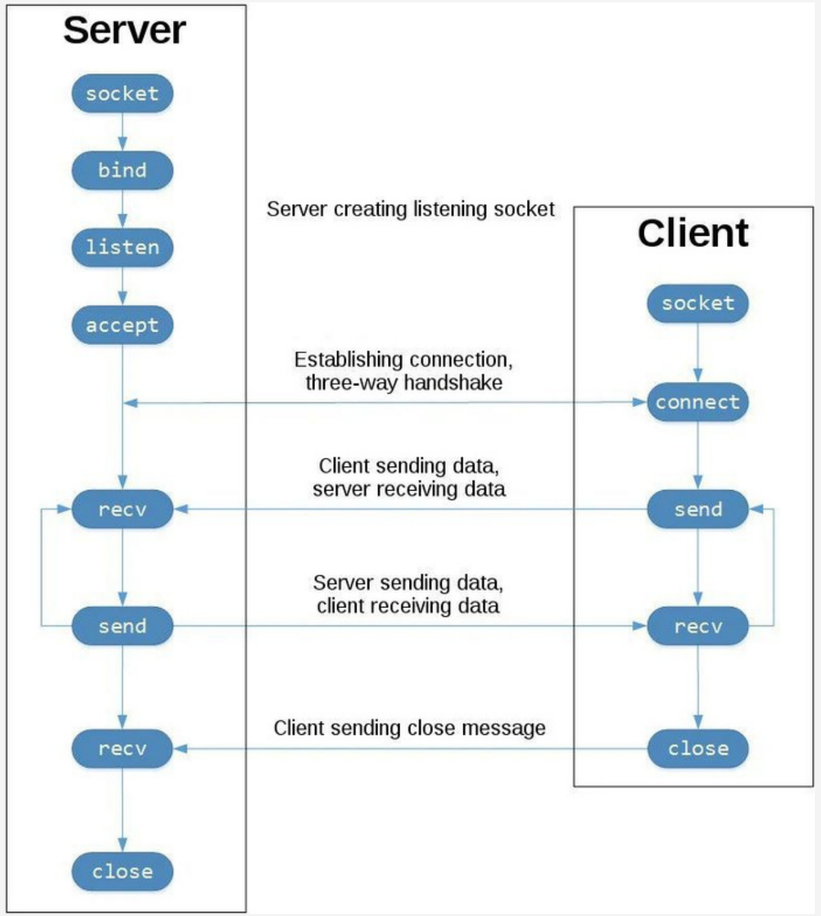
Riesgos y cómo evitarlos

Riesgo	Cómo evitarlo
Certificados autofirmados	*Autoridad certificadora*
Bugs en el script	*Pruebas para identificar los bugs*
Costos altos de servicios en AWS	*Set-Up escalable en AWS*


Crear privkey y cert Central Autenticadora (CA) con OpenSSL
Socket escuchando por protocolo TCP
Pasar CA cert a través de protocolo SSH
Conectar con cliente por TPC/SSL
A partir de este punto scripts de Python
¿Cliente envía o recibe?
Envía
Recibe
Mensaje a enviar: Base de con datos
Mensaje a enviar: Insertar query
Firmado ECDSA de mensaje: se obtiene (r,s,privkey,m) y se envía
Servidor recibe datos y hace verificado
¿Se acepta o se rechaza?
Acepta
Rechaza
Enviar respuesta a cliente
¿Es base de datos o query?
base de datos
query
Subir base de datos a RDS (MySQL)
Extraer datos de RDS
Firmar datos y enviar a cliente
Cliente recibe y hace verificado
Cliente despliega respuesta y si aprobo muestra datos
RDS
Ambos
Servidor EC2
Auditor/Cliente/Rasp.

Anexos


Sender
Reciever
Key Generation:
1
Public
Private
2
Signing Algorithm
Message:
cryptocompare.com
3
Public
Signature
Message
Public
4
Verification Algorithm
5
Message:
Cryptocompare.com
sent by signer


Server
socket
bind
listen
accept
recv
send
recv
close
Server creating listening socket
Client
socket
Establishing connection,
three-way handshake
connect
Client sending data,
server receiving data
send
Server sending data,
client receiving data
recv
Client sending close message
close