End-to-End Security with Arm-Based Edge Devices & IoTeX Blockchain

Xinxin Fan and Raullen Chai
December 3, 2019

The Internet of Trusted Things
Part I: Pebble Tracker Overview
What is Pebble Tracker?

• An “Out-of-the-Box” trusted smart GPS tracker prototyping kit with 4G connectivity and sensor technologies
• Build around the latest low-power nRF9160 System-in-Package (SiP)
• Support LET-M and NB-IoT in bands from 700 MHz to 2.2 GHz
• Support worldwide operation with IoT SIM cards
• Come with environmental and motion sensors for version 1
• Add a light sensor and a buzzer for version 2
• SoC platform security with Arm TrustZone and CryptoCell
• Include an asset tracking and decentralized data authorization demo application
• Open-source development tools and SDK from Nordic
Why Are We Building Pebble Tracker?

Internet of Trusted Things (IoTT)

- Data collection
- Data in transit
- Data at rest
- Data processing
- Data retention

Pebble tracker is an ideal smart device for realizing the IoTT security reference architecture developed by IoTeX

- Secure IoT device
- IoT cloud
- Blockchain

Data Life Cycle for IoTT
Pebble Tracker Hardware

- GPS Antenna
- nRF9160 SiP
- LTE-M/NB-IoT Antenna Connector
- Program/Debug Connector
- TDK ICM-42605 Motion Sensor
- Bosch BME680 Environmental Sensor
- RGB LED
- Power Button
- Micro USB Connector
- FTDI FT232R USB to UART
- Nano/4FF SIM Card Slot
- Reset Button
nRF9160 System-in-Package (SiP)

Key Features

- Full integrated SiP for low-power cellular IoT applications
- Arm Cortex-M33 processor
- Multimode LTE-M/NB-IoT modem with integrated RFFE
- Built-in GPS
- Single variant certified for global operations
- 10x16x1 mm LGA package
Arm Cortex-M33 Processor

- 32-bit processor running at 64 MHz
- Trusted execution environment with Arm TrustZone
- Arm CryptoCell 310 for application layer security
- 1 MB Flash & 256 KB RAM
- 4 x SPI/URAT/TWI
- PDM, I2S, PWM, ADC
- Automated power and clock management
- FPU offering up to 10x performance over software
- MPU for enhanced memory protection
- 32 GPIOs
TrustZone for Armv8-M

Applications:
- User applications
- RTOS
- Device drivers
- Protocol stacks

Normal Resources:
- General peripherals

Isolate secure and non-secure executions
Isolate trusted and non-trusted resources

Secure software:
- Secure boot
- Crypto libraries
- Authentication
- RTOS support APIs

Secure Resources:
- Secure storages
- Crypto accelerators
- TRNG

Normal world
- Non secure app
- Non secure RTOS

Secure world
- Secure app/libs
- Secure RTOS

Control interface
- Security resources
- Asymmetric crypto
- Symmetric crypto
- Data interface

CryptoCell augments TrustZone by providing a range of security subsystems

Root-of-Trust
Always on

PSA Certified

Copyright © 2019 Arm Dev Summit, All Rights Reserved
Arm CryptoCell-310 Security Subsystem

Key Features

- True random number generator (TRNG)
- Pseudorandom number generator (PRNG)
- RSA public key cryptography
- Elliptic curve cryptography (ECC)
- Secure remote password protocol (SRP)
- Hashing functions & HMAC
- AES symmetric encryption
- ChaCha20/Poly1305 symmetric encryption
Cellular Connectivity Module
Low Power LTE-M/NB-IoT Modem + GPS

| Multi-Protocol Low Power LTE | 3GPP LTE release 13 Cat-M1 and Cat-NB1 compliant
|                            | 3GPP LTE release 14 Cat-NB1 and Cat-NB2 compliant
|                            | 770 - 2200 MHz band range |
| LTE Bands                  | 1, 2, 3, 4, 5, 8, 12, 13, 14, 17, 18, 19, 20, 25, 26, 28, 66 |
| Performance                | -108 dBm RX sensitivity (LTE-M) |
|                            | -114 dBm RX sensitivity (NB-IoT) |
|                            | Up to +23 dBm output power |
|                            | 375 kbps (LET-M) and 60 kbps (NB-IoT) |
| Energy Efficiency          | eDRX & PSM |
| Features                   | Built-in GPS |
Sensor Hub

Bosch Sensortec BME680 Environmental Sensor
- Barometric pressure and altitude
- Temperature
- Relative humidity
- Volatile organic compounds (VOCs)

TDK Invensense ICM-42605 Motion Sensor
- 3-axis gyroscope
- 3-axis accelerometer
- Temperature

PMT TSL25721 Ambient Light Sensor
What Can You Measure with Pebble Tracker?

- Location: GPS provides accurate real-time location data, whereas LTE-M/NB-IoT localization techniques using the cell ID method can provide rough location for many uses cases.

- Climate: temperature, humidity, pressure and gas sensors provide insight into a tracking object’s environment.

- Movement: vibration, shock, orientation and light sensors indicate the movement status of a tracking object.
Part II: IoTeX Blockchain Platform Overview
Blockchain In a Nutshell

A blockchain is a record, or ledger, of digital events — one that’s “distributed” between many different parties. It can only be updated by consensus of a majority of the participants in the system. And, once entered, information can never be erased. The blockchain contains a certain and verifiable record of every single transaction ever made.

- **Current System**

  - Central authorities (bank, Fed, notary, escrow, etc.) transfers actual value between two parties
  - Multiple intermediaries and record-keeping are required to facilitate transfer of assets and create trust

- **Blockchain System**

  - Distributed network of computers (nodes) that maintains a shared source of information
  - Transaction data is immutable
  - Peer to Peer transactions using digital tokens to represent assets and value
A Blockchain solution can be initiated as a store of transaction record, but also serve as a platform for further innovation and value extraction.

**Application Description**

- A Blockchain solution can offer automated, high fidelity and low-cost mechanisms for record keeping.
- Core mechanism is the maintenance and modification of a distributed ledger.
- Requires user-specific “keys” – records are kept in the ledger but only accessible by authorized users.

- A Blockchain solution enables secure, near real-time, low-cost transfer of value without an intermediary.
- Records can be transferred to other parties using the decentralized distributed ledger.
- Allows transfer of value between two parties, removing the need for a trusted intermediary.

- A Blockchain solution will transform how contracts are executed.
- Protocol is programmable to trigger transfer of value and information under certain conditions.
- Smart contracts can be developed, exchanged, and automatically executed on decentralized systems.
IoTeX Blockchain - High Throughput

- [https://github.com/iotexproject/iotex-core](https://github.com/iotexproject/iotex-core)
- Blockchain is usually slow (low TPS) and inextensible. How to make a blockchain fast and extensible?
  - Provides high throughput of transactions (3000+ TPS without sharding)
  - Provide consensus-as-a-service for layer2 chains
IoTeX Blockchain - Smart Contract

- User-defined programs running on top of a blockchain
- Smart contract simulates trusted third party with shared state.s
- Applications
  - Assets
  - Lotteries
  - Insurance
  - Supply-chain management
  - Self-sovereign identity
  - And many more!
// A simple bank smart contract where the user
// can deposit and query funds.

pragma solidity ^0.4.0;

// give your contract a name
contract aBank {

    // this declares a key/value array called balanceAccount.
    // the key is address and the value is an unsigned integer
    mapping (address => uint) balanceAccount;

    // declare a deposit function that takes an input called amount
    function deposit(uint amount) public {
        // add amount to the balance of the sender
        balanceAccount[msg.sender] += amount;
    }

    // a getBalance function that accepts no inputs but returns
    // the amount in the balanceAccount array
    function getBalance() returns (uint balance){
        return balanceAccount[msg.sender];
    }

}
IoTeX Blockchain SDK

https://docs.iotex.io/docs/libraries-and-tools.html

`iotex-antenna` is our SDK allowing you to interact with a local or remote iotex blockchain node, using a gRPC or gRPC-web connection.

In your JS project root, use `npm install` or `yarn add`.
```
npm install iotex-antenna
```

In your Golang project root, use `go dep` or `go mod`.
```
go get -u github.com/iotexproject/iotex-antenna-go
```

The embedded SDK will be released soon!
Transfer

antenna.iotx.accounts create, sign, and send a transaction of transfer to iotx blockchain network.

```javascript
import Antenna from "iotex-antenna";
import { toRau } from "iotex-antenna/lib/account/utils";

(async () => {
    const antenna = new Antenna("http://api.testnet.iotx.one:80");
    const unlockedWallet = await antenna.iotx.accounts.privateKeyToAccount("73c7b4e62b1f165dcf9ebdea8270db01ebf5b5630e2ed96b3d2d4889450426");
    const newWallet = antenna.iotx.accounts.create("any entropy");

    const actionHash = await antenna.iotx.sendTransfer(
        from: unlockedWallet.address,
        to: newWallet.address,
        value: toRau("1", "iotx"),
        gasLimit: "100000",
        gasPrice: toRau("1", "Qev")
    );
})();
```

Account

The antenna.iotx.accounts contains functions to generate iotx accounts and sign transactions and data.

```javascript
import Antenna from "iotex-antenna";

(async () => {
    const antenna = new Antenna("http://api.testnet.iotx.one:80");

    const wallet = antenna.iotx.accounts.create();

    const wallet = antenna.iotx.accounts.create();

    // recover the whole wallet from a single private key
    const unlockedWallet = antenna.iotx.accounts.privateKeyToAccount("69805e881e3eadff8fae53d0e58635fb6a6e0fd9e90f6eaad7bc67f3d6c4bd")

    }());
```

arm #ArmDevSummit
RPC Methods

The `rpc-method` (provider) package allows you to make gRPC calls to IoTeX blockchain.

Use the umbrella `antenna.iotx` package:

```javascript
import Antenna from "iotex-antenna";

(async () => {
    const antenna = new Antenna("http://api.testnet.iotex.one:8000");

    const account = await antenna.iotx.getAccount({
        address: "io1cl6rl2ev5dfag5qzmz2x4hafam9vnv2g66q9"});
    const chainMeta = await antenna.iotx.getChainMeta();
    const actions = await antenna.iotx.getActions({
        byIndex: { start: 1, count: 5 },
    });
    const blocks = await antenna.iotx.getBlockMetas({
        byIndex: { start: 1, count: 5 },
    });
})();
```

# XRC20

The XRC20 package provide facility api for XEC20 contract to IoTeX blockchain.

```javascript
import Antenna from "iotex-antenna";
import BigNumber from "bignumber.js";
import { XRC20 } from "iotex-antenna/lib/token/xrc20";

(async () => {
    const antenna = new Antenna("http://api.testnet.iotex.one:8000");

    // init accounts
    const account1 = antenna.iotx.accounts.privateKeyToAccount("privateKey1");
    const account2 = antenna.iotx.accounts.privateKeyToAccount("privateKey2");
    const account3 = antenna.iotx.accounts.privateKeyToAccount("privateKey3");

    // create VITA XRC20 contract instance
    const vita = new XRC20("io1hyw6v7g7mqqyyacfhq67byasnsqj9k0h", { provider: antenna.iotx });

    // token name: IoTeX Vitality
    const name = await vita.name();
```
Welcome to IoTeX Codelabs!

Codelabs provide a guided, tutorial, hands-on coding experience for IoTeX developers. Most codelabs will step you through the process of building a small application, or adding a new feature to an existing application. They cover a wide range of topics such as Node Operation, Smart Contracts, dApps, and more.

Building a rock-paper-scissors DApp
24 min  Updated Aug 20, 2018
Start

Codelab Contribution Guide
11 min  Updated Aug 20, 2019
Start

Deploying a simple smart contract
28 min  Updated Jul 24, 2019
Start

Run a full node on a Raspberry PI
45 min  Updated Oct 24, 2019
Start

Setting up a full node
19 min  Updated Jul 24, 2019
Start
Part III: End-To-End Security Architecture
IoTeX End-To-End Security Architecture Overview
## IoT Endpoint Security

### Hardware Security

- **Device Integrity**
  - Protect device from untrusted S/W
  - Allow device recovery from attacks

- **Resource Protection**
  - Prevent access to certain resources (e.g., key storage)

- **Data Security**
  - Ensure data confidentiality
  - Ensure data integrity

- **Physical Security**
  - Device anti-tampering

### Communication Security

- **Device Authentication**
  - Identities of endpoints and cloud servers

- **Link Encryption**
  - Prevent data eavesdropping

### Device Management

- **Device Management**
  - Device provisioning
  - Device monitoring
  - Remote attestation
  - Firmware update
IoT Cloud Security

- Identity & Access Management
- Data Encryption
- Device Authentication
- Data Authorization
- DDoS Resilience
- Device Management
TIoTA Reference Architecture Integration Patterns

Examples

- 1a: IoT Cloud manages all asset data, selectively writes critical data / events to BC
- 1b: Cloud reads other asset’s data for validation, e.g., for pairing of two assets
- 2a: Track & Trace beacons send critical event to gateway (e.g. via BLE), gateway forwards event to IoT Cloud, data is evaluated and stored in BC
- 4a: On-asset intelligence is executing BC logic and communication directly

Blockchain

BC Middleware

BC Network

IoT Cloud

Gwys/Fog

Field

Asset

1 2 3 4

a) W-Only
b) R-Only
c) RW

1: Asset-to-Cloud-to-BC
2: Gwys-to-Cloud-to-BC
3: Gwys-to-BC
4: Direct

https://www.trusted-iot.org/businesses
What is Digital Identity?

• “The Internet was built without an identity layer” (Kim Cameron: The Laws of Identity, 2006)
• **Digital identity** is a unique representation of a legal entity engaged in an online transaction.
• **Digital identifier** is one or more attributes that uniquely characterize an entity in a specific context.
Three Archetypes of Digital Identity Systems

- **Centralized** (*register once, trusted by one*): A service provider establishes and manages service consumers’ identities and related data in its system.

- **Federated** (*register once, trusted by many*): Service provides trust the identities established and maintained by other identity providers.

- **Decentralized** (*created once, trusted globally*): A service consumer has a self-managed digital identity independent of individual service providers.
Decentralized Identifiers (DIDs)

• Decentralized Identifiers (DIDs) are a new type of identifier for verifiable ‘self-sovereign’ digital identity for individuals, organizations and things.

• The DIDs specification is being developed by the W3C Decentralized Identifier Working Group.

• DIDs have the following important properties:
  – Decentralized: DIDs are designed to function without a central registration authority. DIDs are registered in blockchain or other decentralized network.
  – Cryptographically Verifiable: DIDs are designed to be associated with cryptographic keys and the entities controlling the DID can use those keys to prove ownership.
  – Non-Reassignable: DIDs should be permanent, persistent, and non-reassignable.
  – Resolvable: DIDs are made useful through resolution.
DID Syntax

- The scheme, i.e., did, is fixed
- The method specifies how the DID is created, read (resolved), updated, and deleted (revoked) – CRUD operations.
- The method-specific identifier is an alphanumeric strings that is guaranteed to be unique within the context of the method.

DID Method Registry
A registry for Decentralized Identifier Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>DID Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sovrin</td>
<td>did:sov:</td>
</tr>
<tr>
<td>uPort</td>
<td>did:uport:</td>
</tr>
<tr>
<td>Bitcoin</td>
<td>did:btcr:</td>
</tr>
<tr>
<td>Blockstack</td>
<td>did:stack:</td>
</tr>
<tr>
<td>IPFS</td>
<td>did:ipid:</td>
</tr>
<tr>
<td>IoTeX</td>
<td>did:io:</td>
</tr>
</tbody>
</table>

https://github.com/iotexproject/iotex-did/blob/master/README.md
DID Resolution

- DID Resolution: DID → DID Document
- DID method outlines how to create, retrieve and update the DID on the specific decentralized network (e.g., did:io method for the DID operations on IoTeX blockchain)
- DID infrastructure functions as a global, decentralized key-value store where DIDs act as the keys and DID Documents are the values.
- DID Universal Resolver (https://uniresolver.io)
  - A universal API that works with all DID methods
  - A set of configurable ‘drivers’ that know how to connect to the target DID platform
DID Documents

- DID Document is a JSON-LD document containing six, optional components:
  - The DID that points to the DID Document, identified by the key id
  - A list of public keys identified by the key publicKey
  - List of protocols for authentication control of the DID and delegated capabilities identified by the key authentication
  - A set of service endpoints that allow discovery of way to interact with the entity, identified by the key service
  - A timestamp indicates when the DID Document was created and updated, identified by the key created
  - A digital signature for verifying the integrity of DID Document, identified by the key proof

```json
{  
  "@context": "https://w3id.org/did/v1",  
  "id": "did:sov:123456789abcdefghij",  
  "publicKey": [  
    {  
      "id": "did:sov:123456789abcdefghij#keys-1",  
      "type": "RsaVerificationKey2018",  
      "controller": "did:sov:123456789abcdefghij",  
      "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\n"
    },  
  ],  
  "authentication": [  
    {  
      "type": "RsaSignatureAuthentication2018",  
      "publicKey": "did:sov:123456789abcdefghij#keys-1"
    }]
},  
"service": [  
  {  
    "id": "did:sov:123456789abcdefghij;exam_svc",  
    "type": "ExampleService",  
    "serviceEndpoint": "https://example.com/endpoint/8377464"
  }],  
"created": "2018-02-08T16:03:00Z",
"proof": {  
  "type": "LinkedDataSignature2015",  
  "created": "2018-02-08T16:02:20Z",  
  "creator": "did:sov:8uQhQMGzWxR8vw5P3UWH1ja#keys-1",
  "signatureValue": "QNB13Y7Q9...1tzjn4w=="  
}
}
```
Verifiable claims are statements made by an entity about a ‘subject’ whose authorship can be cryptographically verified

- Government-issued passports enable holders to travel between countries
- Driver’s licenses are used to claim that we are capable of operating a motor vehicle
- University degrees can be used to claim our education status

Ideally, the claims should be cryptographically secure, privacy respecting, and automatically verifiable.
Verifiable Credentials

- Claims are used to create **verifiable credentials** by issuers.
- Verifiable credentials are **decentralized** and **contextual**.
- Credential issuers decide on which claims are contained in the credentials.
- Verifiers **make their own trust decisions** about which credentials to accept.
- Verifiers do not need to contact issuers to perform verification.
- Credential holders are free to choose which credentials to carry and what information to disclose.

```json
{
  "@context": "https://w3id.org/credentials/v1",
  "id": "did:sov:WRfXPg8dantKVubE3HX8pw/credentials/1",
  "type": ["Credential", "NameCredential"],
  "issuer": "did:sov:WRfXPg8dantKVubE3HX8pw",
  "issued": "2019-09-01",
  "claim": {
    "id": "did:btcr:x6lj-wzvr-qqrv-m80w",
    "name": "John Doe",
    "address": "...
  },
  "proof": {
    "type": "RsaSignature2018",
    "created": "2017-06-18T21:19:10Z",
    "creator": "did:sov:WRfXPg8dantKVubE3HX8pw#key-1",
    "nonce": "c0ae1c8e-c7e7-469f-b252-86e6a0e7387e",
    "signatureValue": "BavEll0/I1zpYw8XNi1bgVg/sCneO4Jugez8RWdG/+MCrVpjOboDoe4SxxKjkCOvKCHGtvC4kqj62Jn0Uf9zGFmMATCu5bCB1wPrD+gSutPTLzvueMWMFhWYmflPbbu95t501+rSLHIEuujM+/PXr9Cky6Ed+W3JT24=
  }
}
```
Example - Future Global Supply Chain

- GOV - Government
- CBRA - Cross-Border Regulatory Agency
Decentralized Identifiers of Trusted Things (DITT)

- Facilitate enterprise customers to realize efficient and effective device management and access control across different business units globally.

- Enable consumers to create trusted IoT data marketplace and share services.
Part IV: Pebble Tracker in Action
Decentralized Data Authorization Demo
Tracker Data Collection

- Accept streaming data from tracker with AWS IoT Core
- Persist streaming data to AWS S3 with AWS Kinesis Firehose
Tracker Data Visualization on AWS

- Create a dashboard to visualize GPS coordinates on Google map and sensor data
- Use AWS Kinesis data stream to decouple the processing logic from data ingestion
- Use AWS Lambda function to update the influxdb (time series database) for the new tracker data
- Use Grafana as the time series visualization tool for building dashboard
Device Binding and Verifiable Claim Generation

- Bind user’s blockchain account with the device via a device binding protocol
- Use IoTeX’s account management service (powered by OneFx full-stack framework) to manage a user’s identity and credential
- Use AWS Lambda function to generate a verifiable claim to attest the device ownership
DID Registration with IoTeX Blockchain

- The device owner deploys a service smart contract
- The device owner gets a temporary credential from IoTeX’s account management service
- The device owner uses the temporary credential to directly upload the DID document to AWS S3 and gets the corresponding URI
- The device owner calls the manufacture smart contract to generate DID
Decentralized Tracker Data Authorization
LET'S BUILD DECENTRALIZED FUTURE TOGETHER!