PROGETTO TOO(L)SMART

OUTPUT AZIONE 4 - O.4.a

Codice Output	**O.4.a**
Denominazione	***Implementazione di uno sviluppo con tecnologia blockchain***
Unità di Misura	**Numero**
Valore Target	**1**
Enti coinvolti	Ente Responsabile: UNIME

Descrittivo:

La blockchain costituisce una tecnologia all'avanguardia per la gestione dei dati in maniera distribuita, pertanto il suo utilizzo è stato approfondito all'interno di questo progetto.
Il concetto di base è stato quello di definire e prototipare quanto necessario per certificare i dati raccolti dal sistema, dal momento in cui vengono generati da un sensore a quando vengono immagazzinati nel sistema e resi fruibili agli utenti.
Si è quindi proceduto su due fronti:
1) realizzare un meccanismo tamper-proof sulle schede di acquisizione delle stazioni meteo in grado di garantire l'immutabilità della scheda stessa;
2) realizzare un meccanismo di ledger distribuito basato su blockchain per la memorizzazione dei dati raccolti in grado di assicurare l'immutabilità degli stessi.
Il sistema prototipato è di complemento all'ambiente Stack4Things e costituisce un elemento di valore aggiunto dell'intera architettura. UniMeCoin, invece, non è stata più supportata da UniME. Si è piuttosto preferito concentrare l'attenzione nello sviluppo di meccanismi di validità generale e riutilizzabili basati su blockchain che, in futuro, potrebbero agevolmente essere adottati anche per lo sviluppo di soluzioni di moneta complementare.
Grande attenzione è stata, dunque, dedicata alla trustiness dei dati acquisiti basata su meccanismi di reputazione mediati dalla blockchain, un approfondimento molto utile in quanto collegabile direttamente e immediatamente al verticale ambientale definito nel progetto (e all'affidabilità dei dati provenienti dai sensori, per esempio).

Trustiness dei dati acquisiti basata su meccanismi di reputazione mediati dalla blockchain:
Per dotare i dispositivi IoT degli strumenti necessari a garantire l'autenticità delle informazioni trasmesse è possibile sfruttare sistemi basati su chiavi pubbliche e private attraverso cui è possibile verificare l'autenticità e la non alterazione della fonte e dei dati trasmessi dal dispositivo stesso.
Ogni operazione, attualmente viene usata dal sistema descritto di seguito andrebbe adattata alle esigenze della piattaforma ToolSmart, in quanto al momento il sistema è pensato per fornire trustiness ad un sistema accoppiato ad un diverso sistema di immagazzinamento dati, non idoneo all'interazione con la piattaforma.

I dati grezzi rilevati dal device IoT possono essere rappresentati all'interno del JSON seguente:

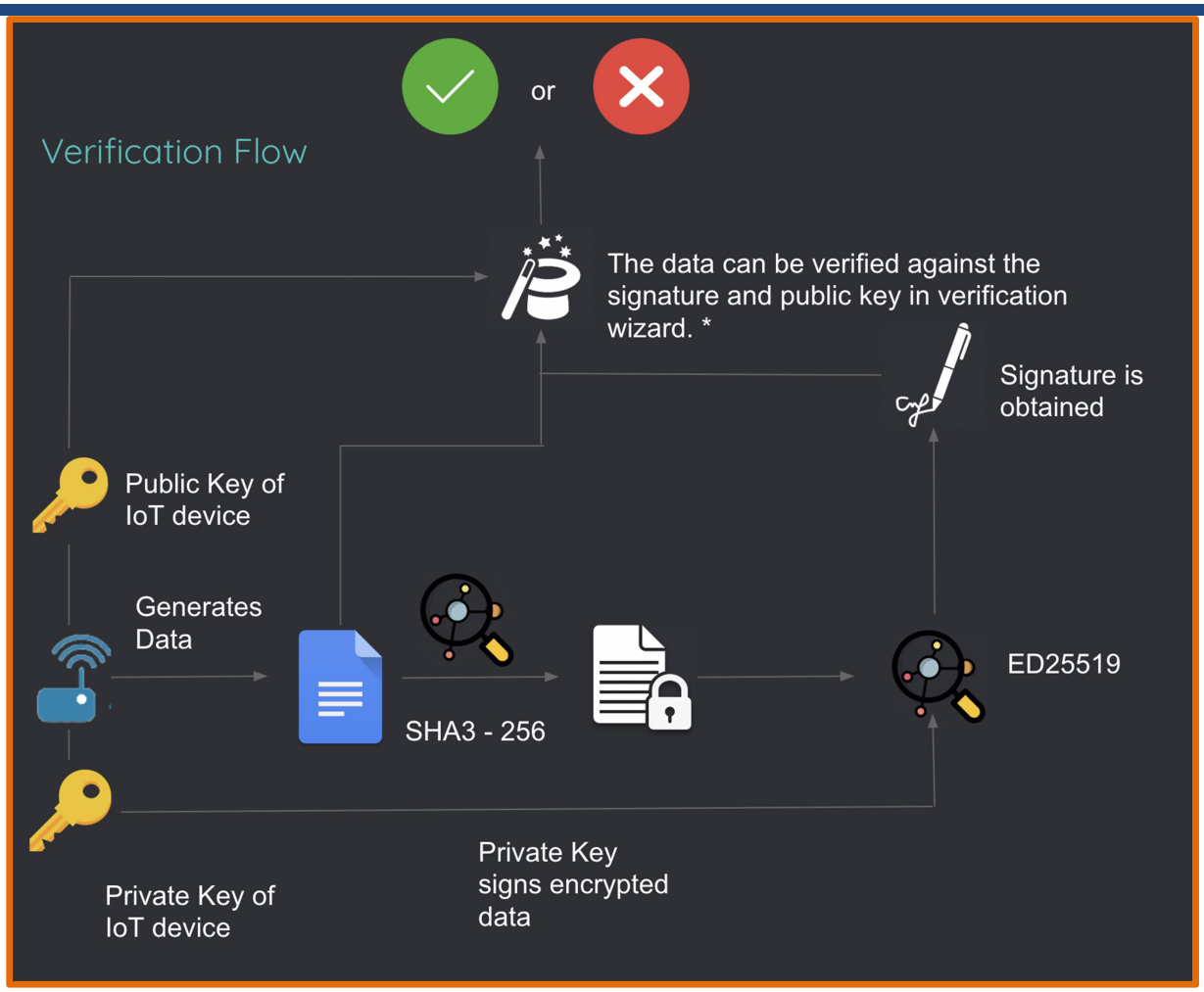

```
{
  "resource_id": "93c39ba9-74cf-4461-b60a-9a206c7fc416",
  "Brightness": "302.5382880655686",
  "Altitude": "19",
```

"Longitude": "15.59541",
"Latitude": "38.25947",
"Date": "2018-07-09T10:20:38.103000",
"entity" : "reading",
"type" : "temperature"
}

Successivamente verranno incapsulati all'interno di un JSON (di cui a breve descriveremo il processo di incapsulamento), e pacchettizzati per il trasferimento attraverso WebSocket.
L'incapsulamento non ha il solo scopo di normalizzare il dato da trasferire al data layer bensì ciò viene fatto affinché si possa far validare i dati dall' IoT layer.
Internamente l'intero set di dati viene convertito in binario, criptato usando SHA-256 e infine firmato con la chiave primaria del device IoT.
Il sistema attualmente in studio, fa sì che il dispositivo IoT crei un JSON secondo la struttura indicata sotto:

{ "_id" : ObjectId("5b339eae6c43d50007d181d5"), "inputs" : [{ "owners_before" : ["3aAQdLVEfBRNif9csvfSmuAfmwhkcYC4GAjU9RikXE3P"], "fulfills" : null, "fulfillment" : "pGSAICY2LW9ZNu0QISifmereGkcuu6awrkKivMe2egL0VflegUDz7u_X8yfE85llyhF2f5da90vhox7_35Kag08VM57X-Uo3yyHpmjBkwp2c34cbqIYYTv6nnYJSvqf06h5RkksF" }], "outputs" : [{ "public_keys" : ["3aAQdLVEfBRNif9csvfSmuAfmwhkcYC4GAjU9RikXE3P"], "condition" : { "details" : { "type" : "ed25519-sha-256", "public_key" : "3aAQdLVEfBRNif9csvfSmuAfmwhkcYC4GAjU9RikXE3P" }, "uri" : "ni:///sha-256;rwKMAXl4xOyYf6P5Ef9O0pFR0Dz6wAOpJ6pXD64f6t0?fpt=ed25519-sha-256&cost=131072" }, "amount" : "1" }], "operation" : "CREATE", "version" : "2.0", "id" : "7abdba2fa828eadcf5179af9f2521668c5a875b894c1dd8bca64bf3e5fe3c749" }

Il JSON sopra indicato rappresenta il messaggio di una transazione realizzata dal device IoT per il prototipo attualmente in analisi, ma nonostante ciò ci permette di individuare alcuni elementi fondamentali come l'id che identifica univocamente l'acquisizione da trasmettere, esso viene generato attraverso il risultato di hashing del JSON e l'oggetto inputs costituito dalla chiave pubblica del device IoT ed un elemento denominato fulfillments contenente la "signature" dei dati trasmessi ad opera del dispositivo.
In fase di verifica dei dati proprio l'analisi delle informazioni fornite dall'oggetto fulfillment ci consentiranno di eseguire la verifica dei dati trasmessi usando l'algoritmo ED25519.
In figura 17, viene mostrato uno schema sintetico del processo di verifica dei dati trasmessi dal dispositivo. Quando i dati devono essere trasmessi dal device IoT i moduli di Lightning-rod leggono sul sensore i valori da trasmettere, li strutturano all'interno di un pacchetto JSON di cui ne ricavano un valore di digest attraverso una funzione SHA3 a 256 bit. Questo digest sarà univoco per il JSON (la garanzia viene dall'algoritmo SHA3) garantendo così l'integrità del pacchetto trasmesso. Il digest inoltre fornirà la possibilità di verificare l'autenticità della sorgente del messaggio, in quanto esso prima della trasmissione verrà cifrato con la chiave privata del device IoT (operazione di firma) garantendo l'identità del device che genera i dati così firmati. In fase di ricezione, i dati ricevuti potranno essere verificati sia in termini di integrità che in termini di autenticità. Partendo dalla verifica di autenticità della sorgente, questa viene realizzata decifrando la firma con la chiave pubblica del device IoT al fine di ottenere il digest relativo ai dati trasmessi. Ottenuto il digest verrà confrontato con quello ottenuto dall'applicazione dell'algoritmo di hashing sul messaggio ricevuto. La verifica del digest verifica conseguentemente anche dell'integrità del contenuto del messaggio, tale verifica infatti, darà esito positivo solo se l'identità del mittente corrisponde e se il messaggio non è stato alterato.



Allegati:

Ulteriori documenti di riferimento relativi agli sviluppi della blockchain sono i seguenti (sono riportati: titolo, publisher e DOI di riferimento, in grassetto i lavori con riferimento specifico a #SmartMe):

- Toward a Trustless Smart City: the #SmartME Experience Publisher: IEEE DOI: 10.1109/WETICE.2019.00051
- Authorization Transparency for Accountable Access to IoT Services Publisher: IEEE DOI: 10.1109/ICIOT.2019.00027
- Blockchain-Based IoT-Cloud Authorization and Delegation Publisher: IEEE DOI: 10.1109/SMARTCOMP.2018.00038
- Blockchain and IoT Integration: A Systematic Survey Publisher: MDPI DOI: https://doi.org/10.3390/s18082575
- Design of a Trustless Smart City system: The #SmartME experiment Publisher: Elsevier DOI: https://doi.org/10.1016/j.iot.2019.100126
- Building a Smart City Service Platform in Messina with the #SmartME Project Publisher: IEEE DOI: 10.1109/WAINA.2018.00109
- Transparent, Provenance-assured, and Secure Software-as-a-Service Publisher: IEEE DOI: 10.1109/NCA.2019.8935014
- An IoT service ecosystem for Smart Cities: The #SmartME project Publisher: Elsevier DOI: https://doi.org/10.1016/j.iot.2018.11.004