NFT TOKEN AND IoT

FIT EUROPE

ethereum

Introduction

Interconnected devices (the IoT) have become the new layer of our environment, almost imprinted into the palpable world.

Due to the proportion (size) of the IoT and its highly heterogeneous nature, vulnerabilities arise at every corner, both at the hardware and at the software level, so the reliability of the deployed devices is highly questionable.

Blockchain technology, which, at the moment, is on the rise, creates the perfect environment for identifying each device in a unique manner, with improved provenance information.

The purpose of our work is to identify a means of tokenization for IoT devices in the context of Ethereum blockchains.

State of the art

Looking at the current market for the Internet of Things devices, the flexibility and diversity are both a blessing and a curse.

Most devices are manufactured in limited-trust environments lacking relevant government regulations, without strong controls. So they are affordable but can become security threats, opening vulnerabilities to the attackers.

With this in mind, imagine building a Smart City (or critical infrastructure), containing billions of devices that can be easily counterfeit or cloned.

The proposed solutions in the literature that have been tackled in many scientific papers use the blockchain technology in order to create a secure management of the IoT devices and NFTs to digitalize them.

For this presentation we have chosen the paper made by Javier Arcenegui et al. for which a reference is given at the end of the presentation.

Non Fungible Token

A Non-Fungible Token (NFT) is a token that use the blockchain to identify something or someone in a unique way.

This type of token is perfect to be used on platforms that offer collectible items, access keys, lottery tickets, numbered seats for concerts and sport matches, etc.

This special type of token has amazing possibilities so it deserves a proper Standard, the ERC-721.

Attributes (ERC-721)

tokenId

unique identifier of the token

owner

owns the token, can transfer ownership and can approve others to act in its name.

Attributes added to the standard

device
Identify address associated with the IoT device

user
Identify BCA address that interacts with IoT devices

Approved and operator
Helps transfer NFT’s to other owners

timestamp
registers if the device checks it is bound with its token.

timeout
Max delay to prove again the bonding

tokenState
Register the actual token state

dataEngagement
Defines the public data needed for agreement

hashK_OD
Register the owner and device secret shared

hashK_UD
Register the user and device secret shared

deviceState
Register the actual device state

Attributes added to the standard

device

Identify address associated with the IoT device

user

Identify BCA address that interacts with IoT devices

Approved and operator

Helps transfer NFT's to other owners

timestamp

Registers if the device checks it is bound with its token.

timeout

Max delay to prove again the bonding

tokenState

Register the actual token state

dataEngagement

Defines the public data needed for agreement

hashK_OD

Register the owner and device secret shared

hashK_UD

Register the user and device secret shared

deviceState

Register the actual device state

Attributes added to the standard

device
Identify address associated with the IoT device

user
Identify BCA address that interacts with IoT devices

Approved and operator
Helps transfer NFT's to other owners

timestamp
Registers if the device checks it is bound with its token.

timeout
Max delay to prove again the bonding

tokenState
Register the actual token state

dataEngagement
Defines the public data needed for agreement

hashK_OD
Register the owner and device secret shared

hashK_UD
Register the user and device secret shared

deviceState
Register the actual device state

Attributes added to the standard

device
Identify address associated with the IoT device

user
Identify BCA address that interacts with IoT devices

Approved and operator
Helps transfer NFT's to other owners

timestamp
Registers if the device checks it is bound with its token.

timeout
Max delay to prove again the bonding

tokenState
Register the actual token state

dataEngagement
Defines the public data needed for agreement

hashK_OD
Register the owner and device secret shared

hashK_UD
Register the user and device secret shared

deviceState
Register the actual device state

Attributes added to the standard

device

Identify address associated with the IoT device

user

Identify BCA address that interacts with IoT devices

Approved and operator

Helps transfer NFT's to other owners

timestamp

Registers if the device checks it is bound with its token.

timeout

Max delay to prove again the bonding

tokenState

Register the actual token state

dataEngagement

Defines the public data needed for agreement

hashK_OD

Register the owner and device secret shared

hashK_UD

Register the user and device secret shared

deviceState

Register the actual device state

Attributes added to the standard

device
Identify address associated with the IoT device

user
Identify BCA address that interacts with IoT devices

Approved and operator
Helps transfer NFT's to other owners

timestamp
Registers if the device checks it is bound with its token.

timeout
Max delay to prove again the bonding

tokenState
Register the actual token state

dataEngagement
Defines the public data needed for agreement

hashK_OD
Register the owner and device secret shared

hashK_UD
Register the user and device secret shared

deviceState
Register the actual device state

Attributes added to the standard

device
Identify address associated with the IoT device

user
Identify BCA address that interacts with IoT devices

Approved and operator
Helps transfer NFT's to other owners

timestamp
Registers if the device checks it is bound with its token.

timeout
Max delay to prove again the bonding

tokenState
Register the actual token state

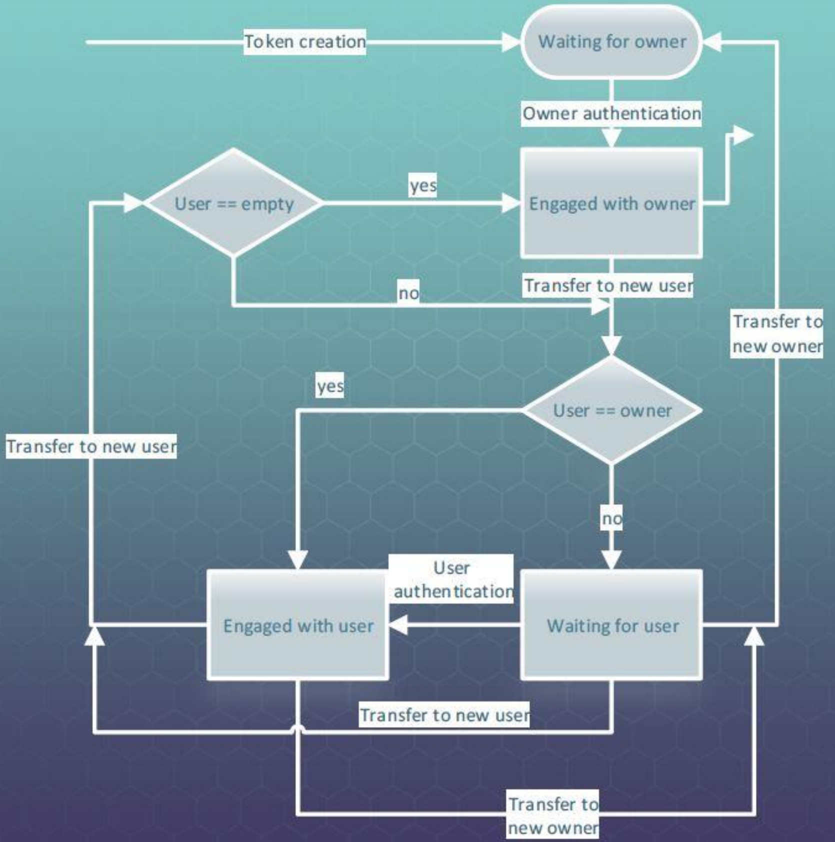
dataEngagement
Defines the public data needed for agreement

hashK_OD
Register the owner and device secret shared

hashK_UD
Register the user and device secret shared

deviceState
Register the actual device state

State Diagram



Attributes added to the standard

device
Identify address associated with the IoT device

user
Identify BCA address that interacts with IoT devices

Approved and operator
Helps transfer NFT's to other owners

timestamp
Registers if the device checks it is bound with its token.

timeout
Max delay to prove again the bonding

tokenState
Register the actual token state

dataEngagement
Defines the public data needed for agreement

hashK_OD
Register the owner and device secret shared

hashK_UD
Register the user and device secret shared

deviceState
Register the actual device state

Attributes added to the standard

device

Identify address associated with the IoT device

user

Identify BCA address that interacts with IoT devices

Approved and operator

Helps transfer NFT's to other owners

timestamp

Registers if the device checks it is bound with its token.

timeout

Max delay to prove again the bonding

tokenState

Register the actual token state

dataEngagement

Defines the public data needed for agreement

hashK_OD

Register the owner and device secret shared

hashK_UD

Register the user and device secret shared

deviceState

Register the actual device state

Attributes added to the standard

device
Identify address associated with the IoT device

user
Identify BCA address that interacts with IoT devices

Approved and operator
Helps transfer NFT's to other owners

timestamp
Registers if the device checks it is bound with its token.

timeout
Max delay to prove again the bonding

tokenState
Register the actual token state

dataEngagement
Defines the public data needed for agreement

hashK_OD
Register the owner and device secret shared

hashK_UD
Register the user and device secret shared

deviceState
Register the actual device state

Attributes added to the standard

device
Identify address associated with the IoT device

user
Identify BCA address that interacts with IoT devices

Approved and operator
Helps transfer NFT's to other owners

timestamp
Registers if the device checks it is bound with its token.

timeout
Max delay to prove again the bonding

tokenState
Register the actual token state

dataEngagement
Defines the public data needed for agreement

hashK_OD
Register the owner and device secret shared

hashK_UD
Register the user and device secret shared

deviceState
Register the actual device state

Functions in the standard

balanceOf
Returns the current balance of the token's owner

ownerOf
Returns the current owner of the token

safeTransferFrom
Transfers the ownership of a token

transferFrom
Transfers the ownership of a token with the possibility of losing the token

isApprovedForAll
Returns if the operator is allowed to manage all of the assets of owner.

approve
Gives permission to operator to transfer token to another account.

getApproved
Returns the account that is approved to manage the token

setApprovalForAll
Approve or remove operator as an operator for the caller.

Functions in the standard

balanceOf
Returns the current balance of the token's owner

ownerOf
Returns the current owner of the token

safeTransferFrom
Transfers the ownership of a token

transferFrom
Transfers the ownership of a token with the possibility of losing the token

isApprovedForAll
Returns if the operator is allowed to manage all of the assets of owner.

approve
Gives permission to operator to transfer token to another account.

getApproved
Returns the account that is approved to manage the token

setApprovalForAll
Approve or remove operator as an operator for the caller.

Functions in the standard

balanceOf
Return the current balance of the token's owner

ownerOf
Returns the current owner of the token

safeTransferFrom
Transfers the ownership of a token

transferFrom
Transfers the ownership of a token with the possibility of losing the token

isApprovedForAll
Returns if the operator is allowed to manage all of the assets of owner.

approve
Gives permission to operator to transfer token to another account.

getApproved
Returns the account that is approved to manage the token

setApprovalForAll
Approve or remove operator as an operator for the caller.

Functions in the standard

balanceOf
Return the current balance of the token's owner

ownerOf
Returns the current owner of the token

safeTransferFrom
Transfers the ownership of a token

transferFrom
Transfers the ownership of a token with the possibility of losing the token

isApprovedForAll
Returns if the operator is allowed to manage all of the assets of owner.

approve
Gives permission to operator to transfer token to another account.

getApproved
Returns the account that is approved to manage the token

setApprovalForAll
Approve or remove operator as an operator for the caller.

Functions in the standard

balanceOf
Return the current balance of the token's owner

ownerOf
Returns the current owner of the token

safeTransferFrom
Transfers the ownership of a token

transferFrom
Transfers the ownership of a token with the possibility of losing the token

isApprovedForAll
Returns if the operator is allowed to manage all of the assets of owner.

approve
Gives permission to operator to transfer token to another account.

getApproved
Returns the account that is approved to manage the token

setApprovalForAll
Approve or remove operator as an operator for the caller.

Functions in the paper

createToken
Creates a New Token Linking a Device BCA Address to a TokenId

startOwner Engagement
Starts Engagement Process between Owner and Device

Owner Engagement
Notifies owner and device of their mutual authentication

startUser Engagement
Starts engagement process between user and device

userEngagement
Notifies user and device of their mutual authentication

setUser
The owner assigns a user to the token

Update Timestamp
The device updates the attribute timestamp of its token

setTimeout
The owner of the token sets its attribute timeout

checkTimeout
Checks if the device remains bound to its token

Functions in the paper

createToken
Creates a New Token Linking a Device BCA Address to a TokenId

startOwner Engagement
Starts Engagement Process between Owner and Device

Owner Engagement
Notifies owner and device of their mutual authentication

startUser Engagement
Starts engagement process between user and device

userEngagement
Notifies user and device of their mutual authentication

setUser
The owner assigns a user to the token

Update Timestamp
The device updates the attribute timestamp of its token

setTimeout
The owner of the token sets its attribute timeout

checkTimeout
Checks if the device remains bound to its token

Functions in the paper

createToken
Creates a New Token Linking a Device BCA Address to a TokenId

startOwner Engagement
Starts Engagement Process between Owner and Device

Owner Engagement
Notifies owner and device of their mutual authentication

startUser Engagement
Starts engagement process between user and device

userEngagement
Notifies user and device of their mutual authentication

setUser
The owner assigns a user to the token

Update Timestamp
The device updates the attribute timestamp of its token

setTimeout
The owner of the token sets its attribute timeout

checkTimeout
Checks if the device remains bound to its token

Functions in the paper

createToken
Creates a New Token Linking a Device BCA Address to a TokenId

startOwner Engagement
Starts Engagement Process between Owner and Device

Owner Engagement
Notifies owner and device of their mutual authentication

startUser Engagement
Starts engagement process between user and device

userEngagement
Notifies user and device of their mutual authentication

setUser
The owner assigns a user to the token

Update Timestamp
The device updates the attribute timestamp of its token

setTimeout
The owner of the token sets its attribute timeout

checkTimeout
Checks if the device remains bound to its token

Functions in the paper

createToken

Creates a New Token Linking a Device BCA Address to a TokenId

startOwner Engagement

Starts Engagement Process between Owner and Device

Owner Engagement

Notifies owner and device of their mutual authentication

startUser Engagement

Starts engagement process between user and device

userEngagement

Notifies user and device of their mutual authentication

setUser

The owner assigns a user to the token

Update Timestamp

The device updates the attribute timestamp of its token

setTimeout

The owner of the token sets its attribute timeout

checkTimeout

Checks if the device remains bound to its token

Functions in the paper

createToken
Creates a New Token Linking a Device BCA Address to a TokenId

startOwner Engagement
Starts Engagement Process between Owner and Device

Owner Engagement
Notifies owner and device of their mutual authentication

startUser Engagement
Starts engagement process between user and device

userEngagement
Notifies user and device of their mutual authentication

setUser
The owner assigns a user to the token

Update Timestamp
The device updates the attribute timestamp of its token

setTimeout
The owner of the token sets its attribute timeout

checkTimeout
Checks if the device remains bound to its token

Functions in the paper

createToken
Creates a New Token Linking a Device BCA Address to a TokenId

startOwner Engagement
Starts Engagement Process between Owner and Device

Owner Engagement
Notifies owner and device of their mutual authentication

startUser Engagement
Starts engagement process between user and device

userEngagement
Notifies user and device of their mutual authentication

setUser
The owner assigns a user to the token

Update Timestamp
The device updates the attribute timestamp of its token

setTimeout
The owner of the token sets its attribute timeout

checkTimeout
Checks if the device remains bound to its token

Functions we added

checkTheLink

Sends a message to the device. If we get any response back, the link is still active, if not it's disconnected

setDeviceState

Sets the device state

getDeviceState

Gets the device state

Functions we added

checkTheLink

Sends a message to the device. If we get any response back, the link is still active, if not it's disconnected

setDeviceState

Sets the device state

getDeviceState

Gets the device state

Functions we added

checkTheLink

Sends a message to the device. If we get any response back, the link is still active, if not it's disconnected

setDeviceState

Sets the device state

getDeviceState

Gets the device state

Functions we added

checkTheLink

Sends a message to the device. If we get any response back, the link is still active, if not it's disconnected

setDeviceState

Sets the device state

getDeviceState

Gets the device state

Events in the standard

Transfer

Emitted when tokenId token is transferred from owner to someone new.

Approval

Emitted when owner enables someone to manage the tokenId token.

ApprovalForAll

Emitted when owner enables or disables someone to manage all of its assets.

Events in the standard

Transfer

Emitted when tokenId token is transferred from owner to someone new.

Approval

Emitted when owner enables someone to manage the tokenId token.

ApprovalForAll

Emitted when owner enables or disables someone to manage all of its assets.

Events in the standard

Transfer

Emitted when tokenId token is transferred from owner to someone new.

Approval

Emitted when owner enables someone to manage the tokenId token.

ApprovalForAll

Emitted when owner enables or disables someone to manage all of its assets.

Events in the standard

Transfer

Emitted when tokenId token is transferred from owner to someone new.

Approval

Emitted when owner enables someone to manage the tokenId token.

ApprovalForAll

Emitted when owner enables or disables someone to manage all of its assets.

Events in the paper

OwnerEngaged
Emitted when owner and device have Engaged.

UserAssigned
Emitted when a new user is set.

UserEngaged
Emitted when user and device have Engaged.

TimeoutAlarm
Emitted when timeout is passed

Events in the paper

OwnerEngaged

Emitted when owner and device have Engaged.

UserAssigned

Emitted when a new user is set.

UserEngaged

Emitted when user and device have Engaged.

TimeoutAlarm

Emitted when timeout is passed

Events in the paper

OwnerEngaged

Emitted when owner and device have Engaged.

UserAssigned

Emitted when a new user is set.

UserEngaged

Emitted when user and device have Engaged.

TimeoutAlarm

Emitted when timeout is passed

Events in the paper

OwnerEngaged

Emitted when owner and device have Engaged.

UserAssigned

Emitted when a new user is set.

UserEngaged

Emitted when user and device have Engaged.

TimeoutAlarm

Emitted when timeout is passed

Events in the paper

OwnerEngaged

Emitted when owner and device have Engaged.

UserAssigned

Emitted when a new user is set.

UserEngaged

Emitted when user and device have Engaged.

TimeoutAlarm

Emitted when timeout is passed

Event we added

DeviceMalfunctioned

Emitted when the link is disconnected

Event we added

DeviceMalfunctioned

Emitted when the link is disconnected

Some Assumptions

1

Trusted Manufacturer

We assume that the Manufacturer creates an unique TokenID and puts it into the ROM of the device SoC

2

Non-leakage SoC

We assume that the Soc has no leakage so all the SoC internal components are trusted.

3

Secure Boot

We assume that a secure boot process is included in the IoT devices.

?

More Trust on Manufacturer

Proposal of Physically Binding IoT Devices

Private Key

Manufacturer puts in the ROM of the SoC of the IoT devices the TokenID and his PKman

Hide keys

The TokenID is obfuscated and reconstructed with physical unclonable functions (PUFs)

Sign SW

All the Firmware and Software in the device is signed using the TokenID so we can avoid malicious modifications

Root of Trust

Zero-Stage Bootloader (ZSB), is located in the SoC OTP memory. This is our “Root of Trust”

Secure Boot

A secure boot process is performed, in order to verify the firmware and software.

Trust ??

The manufacturer creates the NFT Token in the blockchain

Conclusions

- Binding IoT devices with an NFT is possible
- We explained how to extend the ERC-721
- A lot of trust in the manufacturer is required
- Research needs to be continued on this topic

Future work

- A future work could be to detach the trust between the manufacturer and the device so that the owner can create and add a device to the blockchain so that no trust in the manufacturer will be required anymore.

who are we ?

$%&fs g$f92!

$%&/j g$f62)

!dft556 /j5£d4

0GDg3& $fs/(£

£htg97

Yellow Team

ANDREA

SIMONE

NEGIN

MADALINA

ROMAIN

NICOLAS

GULSAH

SILVIA

SAMUEL

References

> https://www.mdpi.com/1424-8220/21/9/3119

> https://link.springer.com/chapter/10.1007/978-3-030-61638-0_2

> https://blockchain.pwias.ubc.ca/sites/blockchain.pwias.ubc.ca/files/report-files/Weingaertner_Tokenization_IoT_AI%20(1).pdf

> https://harborresearch.com/more-than-a-meme-nfts-and-the-iot/

> https://www.sciencedirect.com/science/article/abs/pii/S0167739X19317686

> https://www.researchgate.net/profile/Jonas-Gross-2/publication/344275773_Convergence_of_Blockchain_IoT_and_AI/links/5f69b997458515b7cf46b4e4/Convergence-of-Blockchain-IoT-and-AI.pdf

> https://ieeexplore.ieee.org/document/8726523

QUESTIONS

Instructions for use (premium users)

Infographics

You can add and edit some **infographics** to your presentation to show your data in a visual way.

- Choose your favourite infographic and insert it in your presentation using Ctrl C + Ctrl V or Cmd C + Cmd V in Mac.
- Select one of the parts and **ungroup** it by right-clicking and choosing "Ungroup".
- **Change the color** by clicking on the paint bucket.
- Then **resize** the element by clicking and dragging one of the square-shaped points of its bounding box (the cursor should look like a double-headed arrow). Remember to hold Shift while dragging to keep the proportions.
- **Group** the elements again by selecting them, right-clicking and choosing "Group".
- Repeat the steps above with the other parts and when you're done editing, copy the end result and paste it into your presentation.
- Remember to choose the **"Keep source formatting"** option so that it keeps the design. For more info, please visit **Slidesgo School**.