Department of Computer Science,
National Tsing Hua University,
Taiwan

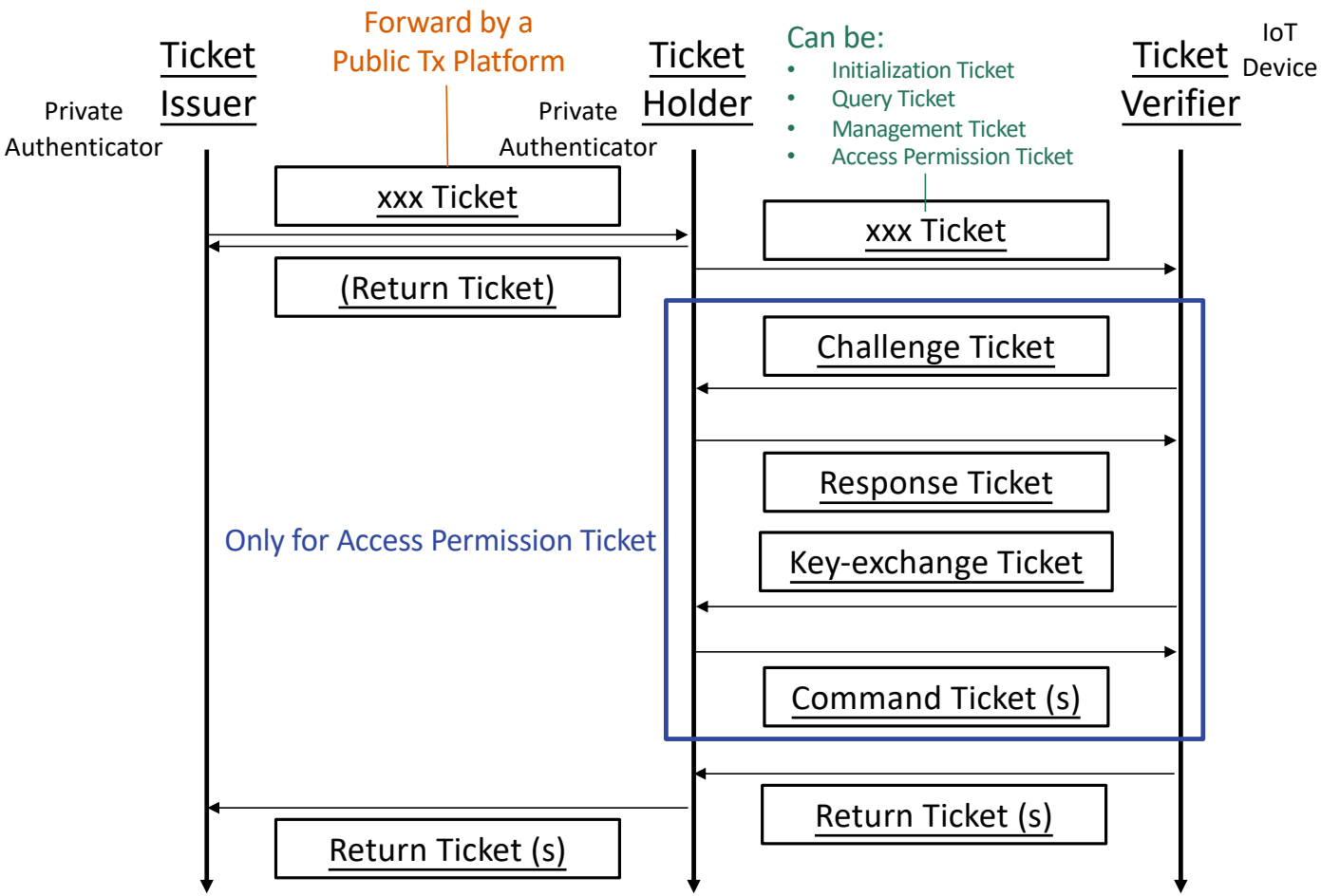
Ureka Ticket System: Development Doc

Technical Introduction Document

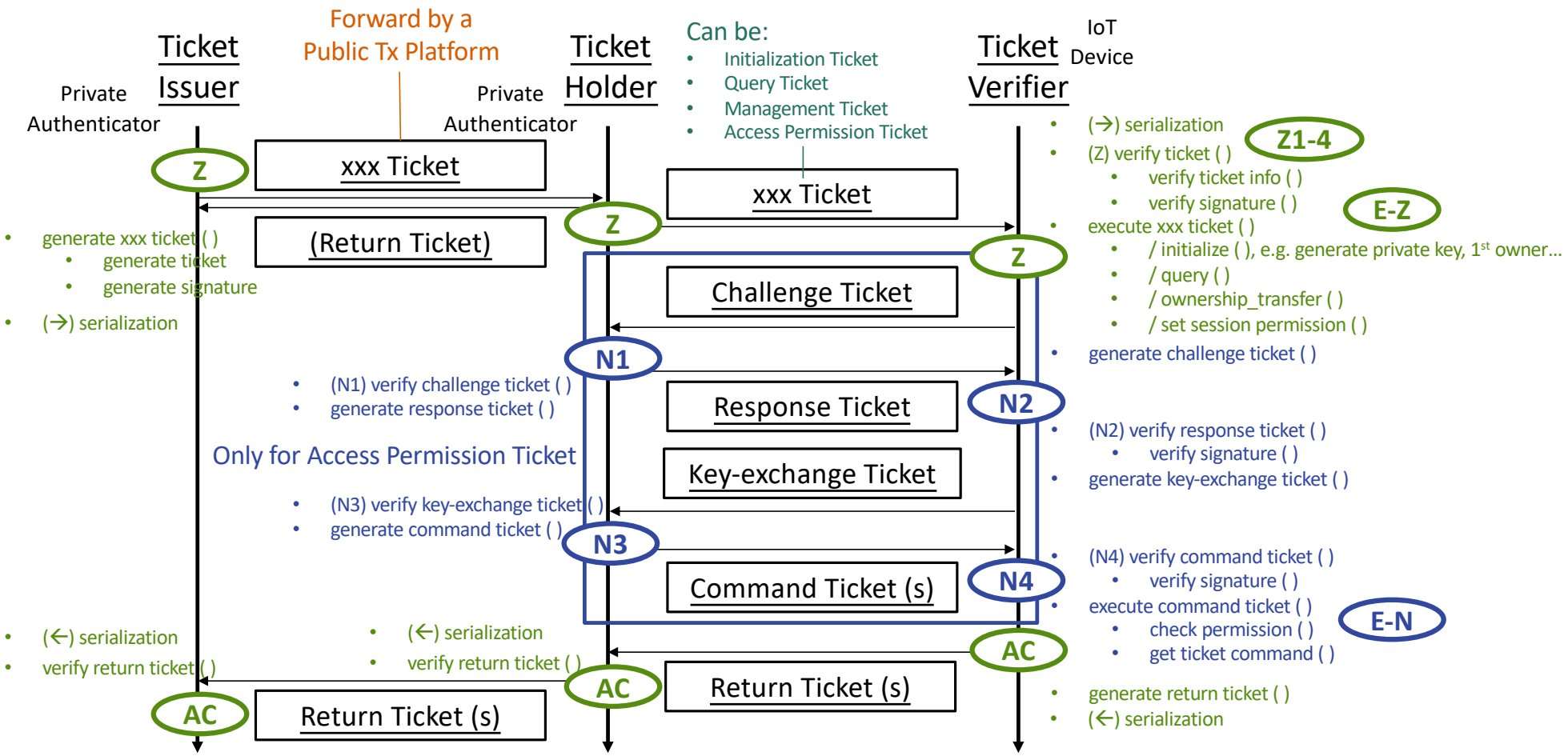
written by
Ken Yang, Logos Lab

09 25, 2020

Protocol: **TICKET HANDSHAKE**



Development Details: **TICKET HANDSHAKE**



Protocol: **TICKET FORMAT**

Device Identity

- Public Key

Ticket Holder Identity

- Public Key

Ticket Issuer Identity

- Public Key

Signature of Ticket Issuer

- (Signature Algorithm)
- Signature Value

Ticket Protocol Version

(I) Owner's consent
[Owner-signed Certificate]

- Ureka Ticket Version
  (protocol maintenance)

Ticket Type

(II) Detailed contents
[Ownership / IoT Data & Control]

- [ ] Initialization Ticket
- [ ] Query Ticket
- [ ] Management Ticket
- [ ] Access Permission Ticket
  - With Challenge, Response & Key-exchange Ticket
- [ ] Command Ticket
- ----------
- [ ] Return Ticket
- ----------

Request Body

- Depend on different ticket types

Response Body

- Only used in Return Ticket

Ticket Type	Description
Initialization	Used to initialize IoT device, generate its private key & set its 1st owner.
Query	Used to get IoT device's status, for example, used to know IoT device's Id, or used to know IoT device's owner Id.
Management	Used to change IoT device's owner
Access Permission	Do not change owner, but let owner or authorized user access IoT device with specific permissions With user authentication (Challenge-response) & session key generation procedures.
Command	Used to specify user's command, where only the commands included in the access permission are legal.
Return	Every Ticket will respond a Return Ticket, which is signed by IoT device, and used to record some response messages, some response data, or even some error messages.

generate/verify_xxx_ticket ( ) add/verify_signature_on_ticket ( )

Z-1 Z-2 Z-3 Z-4

Protocol Version	Ticket Type	Device Identity	Issuer Identity + Signature	Holder Identity	Request Body	Response Body (Return Ticket only)
Ureka-1.0	Initialization [初始設定票券]	-	(No Sig.)	initialize ( ) / 1st Owner	{ DEVICE-TYPE: / PRIVATE-AUTHENTICATOR / IOT-DEVICE }	{ Request Ticket Type, Device Id, Device Status (Success/Fail) }
	Query [狀態詢問票券]	-	-	-	Query-able / Non-Query-able	{ Request Ticket Type, Device Id (or Owner Id), Device Status }
	Management [管理權設定票券]	Device	Owner Sig.	ownership_transfer ( ) / New Owner	{ MANAGEMENT-TYPE: New Owner}	{ Request Ticket Type, Device Status (Success/Fail) }
	Access Permission (+ Challenge, Response & Key-exchange Ticket) [服務權限設定票券]	Device	Owner Sig.	set_session_permission ( ) / Owner / New User (s)	{ RESOURCE-TREE: / Owner Permission (all) / New User Permission (s) (in resource tree) }	{ Request Ticket Type, Device Status (Success/Fail) (user cache may exist) }
	Command [服務指令票券]	Device	/ Owner Signature / User Signature	-	{ [Encrypted] Command }	{ Request Ticket Type, (Success/Fail), [Encrypted] Data }
	Return [狀態回報票券]	(Same as original Request Ticket)	Device Signature	(Same as original Request Ticket)		(As above)

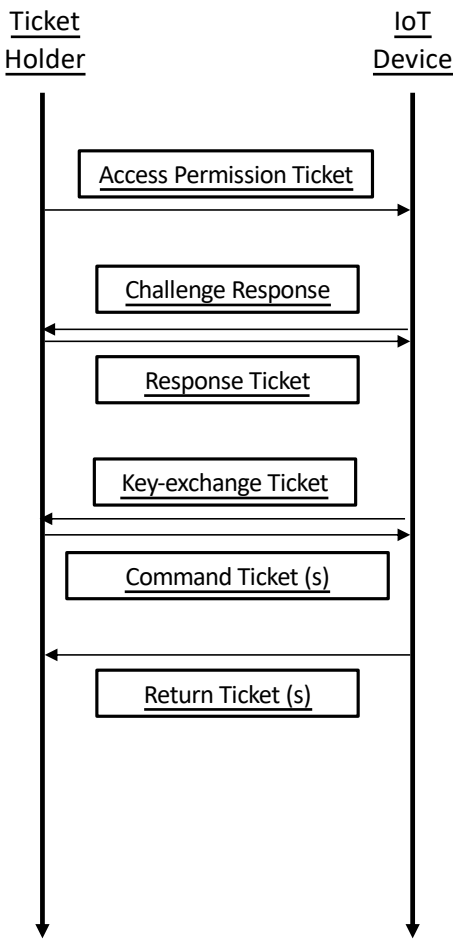
E-Z E-Z E-Z

N1-4 add/verify_signature_on_ challenge_ticket ( ) / response_ticket ( ) / key_exchange_ticket ( )

check_permission ( ) verify_command_ticket ( ) E-N

generate/verify_xxx_ticket ( ) add/verify_signature_on_ticket ( ) AC

Basic Access Protocol



Protocol Version	Ticket Type	Device Identity	Issuer Identity + Signature	Holder Identity set_session_permission ( )	Request Body (->) / Request Body (<-)
Ureka-1.0	(->) Access Permission	Device	Owner Sig. Z-4	/ Owner / New User	E-Z { Resource-tree: / Owner Permission (all) / New User Permission (s) (in resource tree) }
	(<-) Challenge	Device	Device Sig. (i.e. like a Return Ticket) N1	/ Owner / New User	{ Challenge (Random Number) }
	(->) Response	Device	Owner Sig. / New User Sig. (i.e. Holder's Response) N2	(omitted)	
	(<-) Key-exchange	Device	Device Sig. (i.e. like a Return Ticket) N3	/ Owner / New User	{ Salt (Random Number) }
	(->) Command (s)	Device	Owner Sig. / New User Sig. N4	/ Owner / New User check_permission ( ) verify_command_ticket ( )	{ [Encrypted] Command } E-N
	(<->) AES				