GridLogic 2FA: Two-Factor Authentication for Critical Infrastructure Commands in the Field

Kareem Ahmad, **Arman Allahverdi**, Vincent John Mooney III, and Santiago Grijalva
Georgia Institute Of Technology

TPEC 2026

2017-2026 10

Acknowledgement

This work has been partially supported by the U.S. Department of Energy (DoE) Office of Cybersecurity, Energy Security, and Emergency Response (CESER) under Cybersecurity for Energy Delivery Systems (CEDS) Agreement Number #DE-CR0000055 to the Georgia Tech Research Corporation: GridLogic: Hardware/Software Codesign for Deep Grid Visibility and Security

Outline

1. Problem Statement
2. System Architecture
3. Threat Model
4. 2FA Protocol
5. Experiments
    1. Security Experiments
    2. Performance Experiments
6. Discussion
7. References

Outline

1. **Problem Statement**
2. System Architecture
3. Threat Model
4. 2FA Protocol
5. Experiments
    1. Security Experiments
    2. Performance Experiments
6. Discussion
7. References

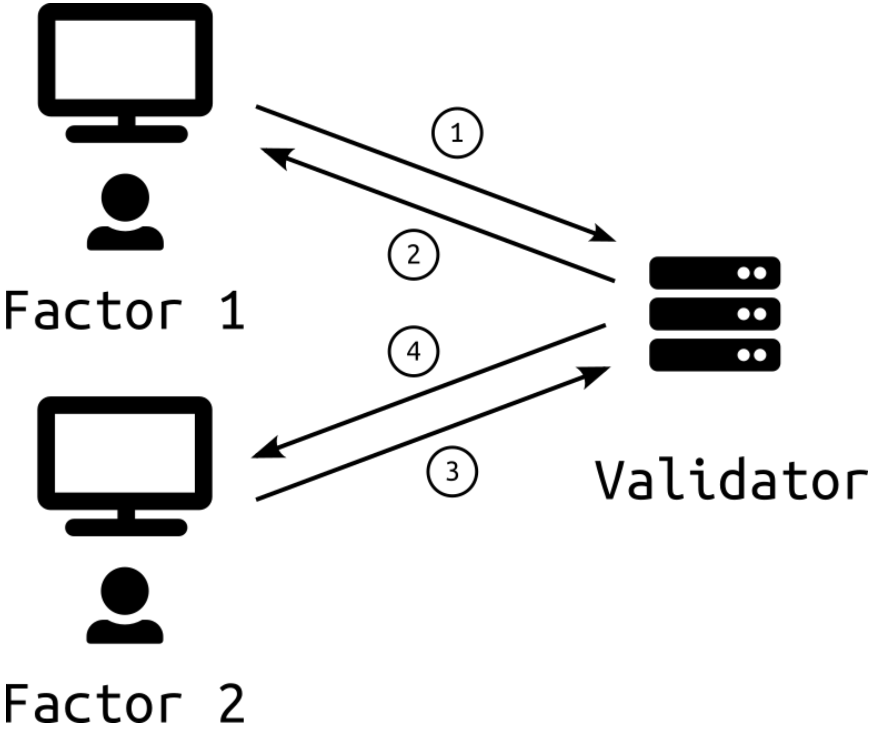
Problem Statement

- Power utilities are a critical component of modern infrastructure
- Cyberattack rates against utilities are increasing rapidly
  - 42% year over year [2]
  - 1157 per week globally in 2025 [2]
- Many utilities disable encryption in their SCADA networks due to overhead concerns or legacy system support

Photo from collaboration with Marrietta Power

Goals

- Develop a Two-Factor Authentication (2FA) protocol that can intercept malicious commands before they are executed
- Empirically measure the overhead of enabling SSL in a test subsystem with and without 2FA

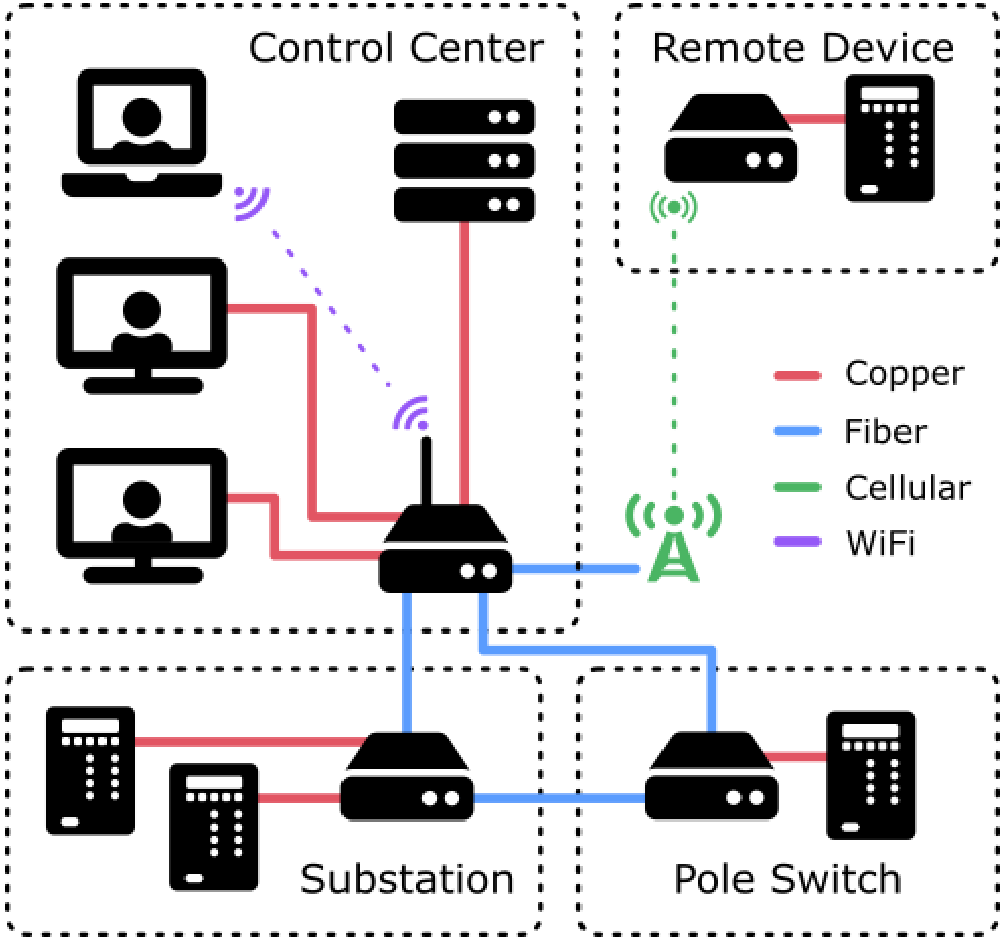


Outline

1. Problem Statement
2. **System Architecture**
3. Threat Model
4. 2FA Protocol
5. Experiments
    1. Security Experiments
    2. Performance Experiments
6. Discussion
7. References

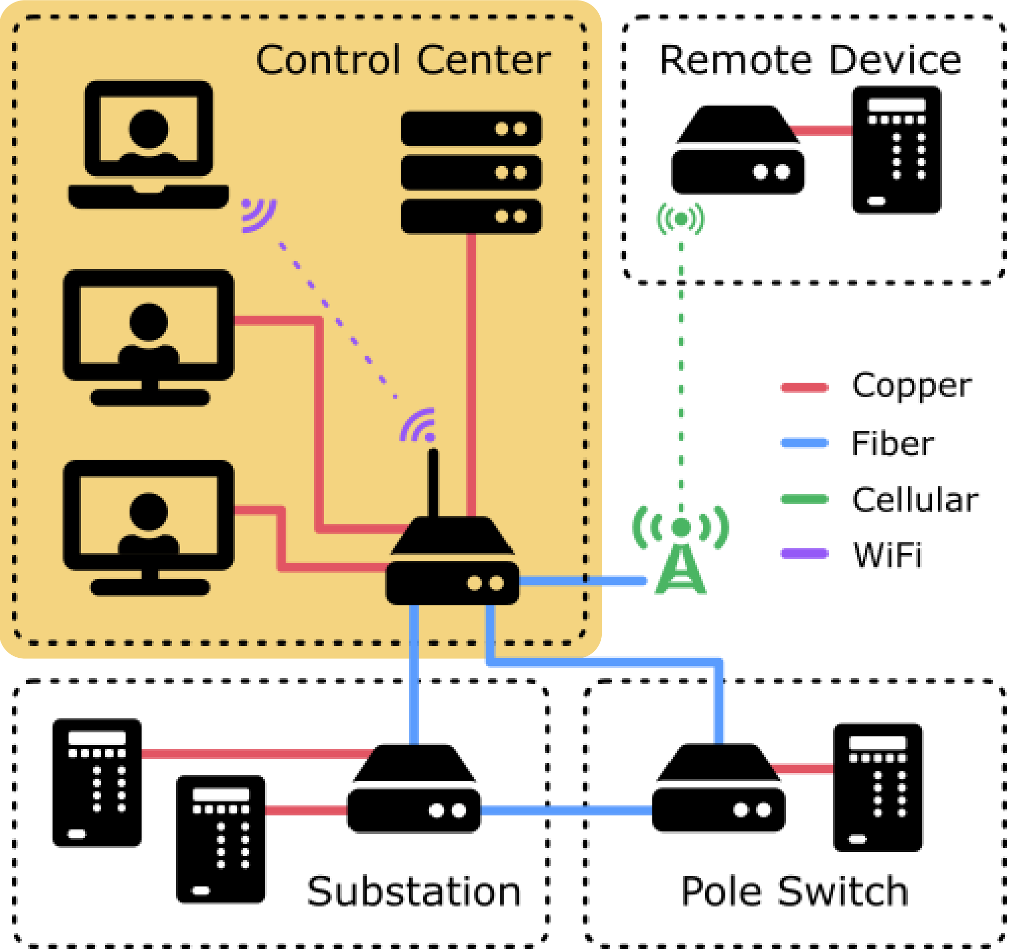
System Architecture

- **Control Center**: Location in a utility where servers, engineers and grid operators are co-located
- **Field Device**: Power-system control device (e.g., digital relay)
- **Interfacing Device**: A device that adds GridLogic 2FA functionality to a Non-GridLogic Field Device
- **GridLogic Device**: A Field Device that supports the GridLogic 2FA protocol, or a Interfacing Device in front of a Non-GridLogic Field Device
- **Issuer**: Entity that generates and sends commands.
  - May be untrusted (e.g., a lone-wolf insider grid operator or a remote attacker)
- **Validator**: Entity that signs commands and coordinates 2FA
- **Authorizer**: Entity that can approve or deny commands. The second factor in 2FA
  - The authorizer must be trusted, typically a senior engineer or manager
- **GridLogic 2FA Protocol**
  - A protocol that coordinates Field Devices, Issuers, Authorizers, with the Validator and Decision Engine to enable command interception and intervention.



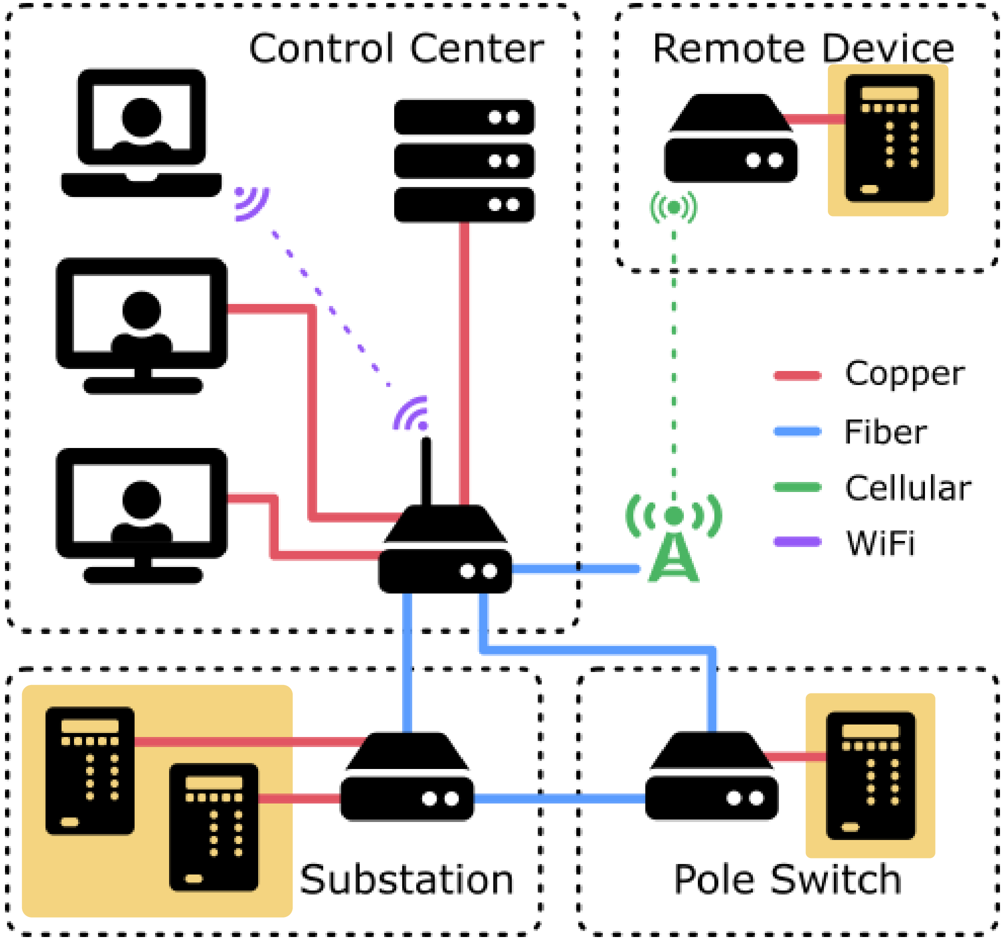
System Architecture

- **Control Center**: Location in a utility where servers, engineers and grid operators are co-located
- **Field Device**: Power-system control device (e.g., digital relay)
- **Interfacing Device**: A device that adds GridLogic 2FA functionality to a Non-GridLogic Field Device
- **GridLogic Device**: A Field Device that supports the GridLogic 2FA protocol, or a Interfacing Device in front of a Non-GridLogic Field Device
- **Issuer**: Entity that generates and sends commands.
  - May be untrusted (e.g., a lone-wolf insider grid operator or a remote attacker)
- **Validator**: Entity that signs commands and coordinates 2FA
- **Authorizer**: Entity that can approve or deny commands. The second factor in 2FA
  - The authorizer must be trusted, typically a senior engineer or manager
- **GridLogic 2FA Protocol**
  - A protocol that coordinates Field Devices, Issuers, Authorizers, with the Validator and Decision Engine to enable command interception and intervention.



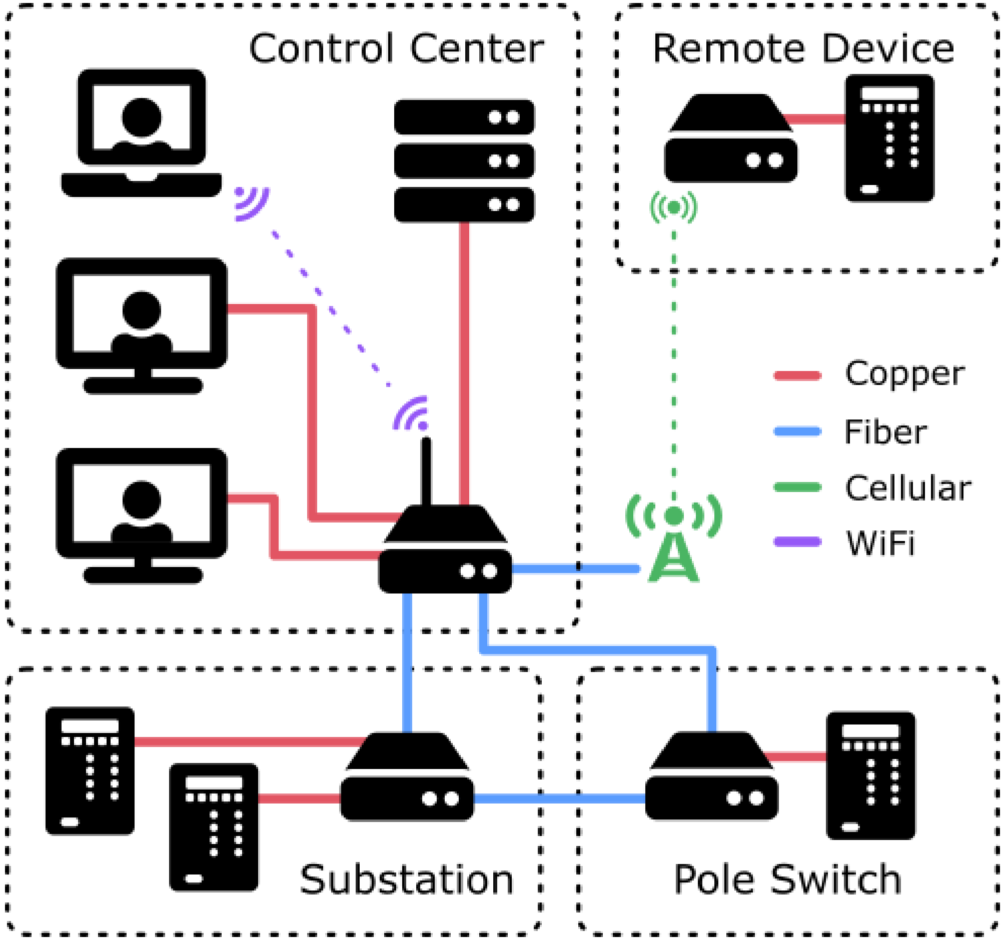
System Architecture

- **Control Center**: Location in a utility where servers, engineers and grid operators are co-located
- **Field Device**: Power-system control device (e.g., digital relay)
- **Interfacing Device**: A device that adds GridLogic 2FA functionality to a Non-GridLogic Field Device
- **GridLogic Device**: A Field Device that supports the GridLogic 2FA protocol, or a Interfacing Device in front of a Non-GridLogic Field Device
- **Issuer**: Entity that generates and sends commands.
  - May be untrusted (e.g., a lone-wolf insider grid operator or a remote attacker)
- **Validator**: Entity that signs commands and coordinates 2FA
- **Authorizer**: Entity that can approve or deny commands. The second factor in 2FA
  - The authorizer must be trusted, typically a senior engineer or manager
- **GridLogic 2FA Protocol**
  - A protocol that coordinates Field Devices, Issuers, Authorizers, with the Validator and Decision Engine to enable command interception and intervention.



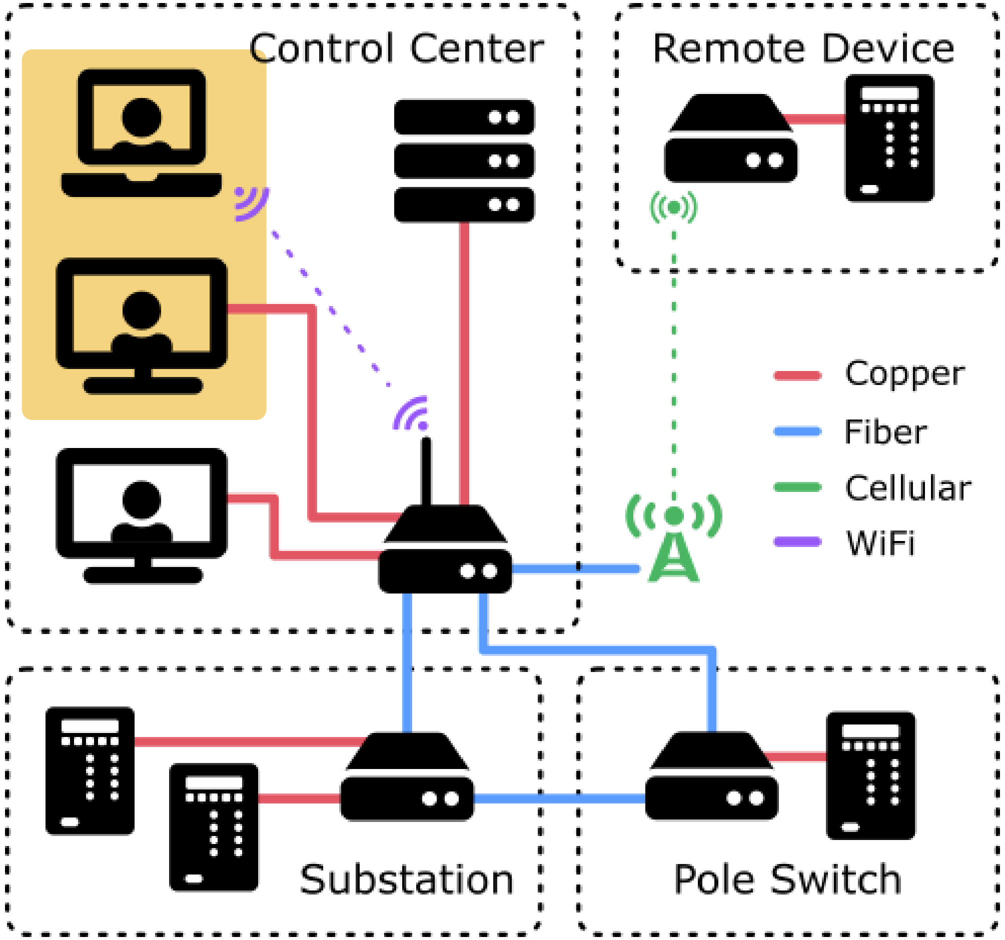
System Architecture

- **Control Center**: Location in a utility where servers, engineers and grid operators are co-located
- **Field Device**: Power-system control device (e.g., digital relay)
- **Interfacing Device**: A device that adds GridLogic 2FA functionality to a Non-GridLogic Field Device
- **GridLogic Device**: A Field Device that supports the GridLogic 2FA protocol, or a Interfacing Device in front of a Non-GridLogic Field Device
- **Issuer**: Entity that generates and sends commands.
  - May be untrusted (e.g., a lone-wolf insider grid operator or a remote attacker)
- **Validator**: Entity that signs commands and coordinates 2FA
- **Authorizer**: Entity that can approve or deny commands. The second factor in 2FA
  - The authorizer must be trusted, typically a senior engineer or manager
- **GridLogic 2FA Protocol**
  - A protocol that coordinates Field Devices, Issuers, Authorizers, with the Validator and Decision Engine to enable command interception and intervention.



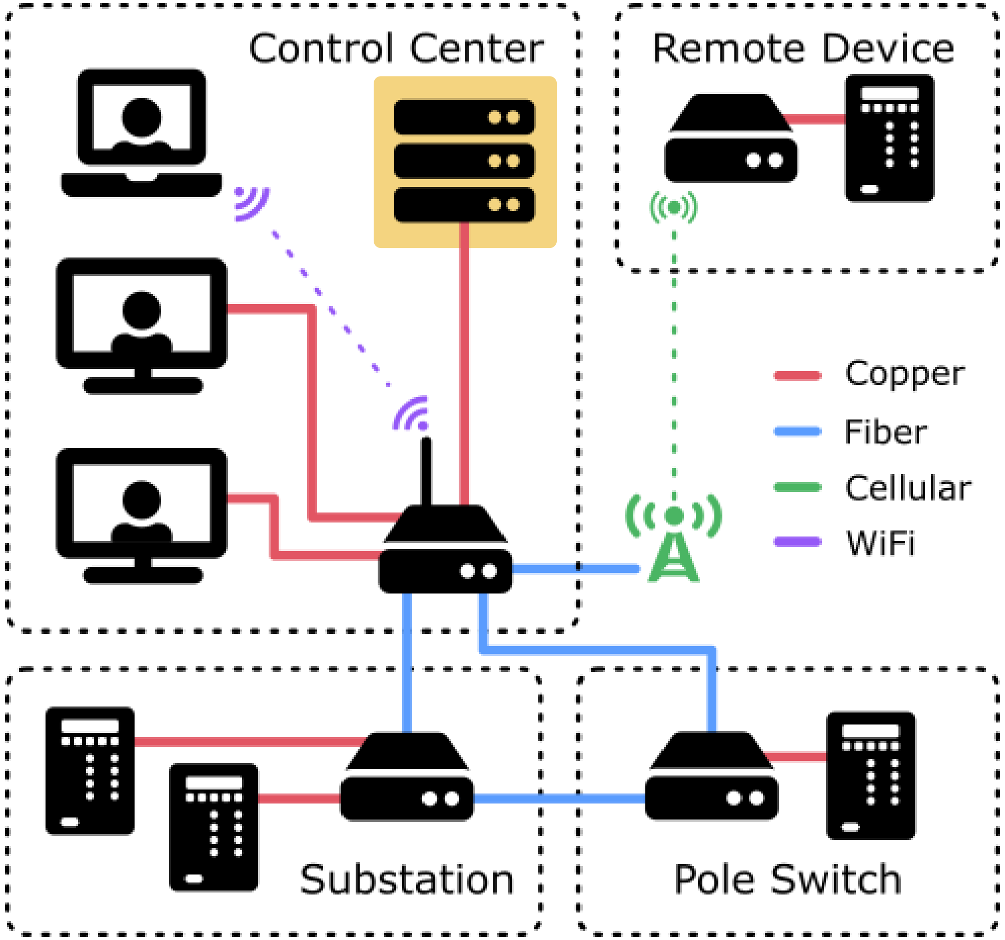
System Architecture

- **Control Center**: Location in a utility where servers, engineers and grid operators are co-located
- **Field Device**: Power-system control device (e.g., digital relay)
- **Interfacing Device**: A device that adds GridLogic 2FA functionality to a Non-GridLogic Field Device
- **GridLogic Device**: A Field Device that supports the GridLogic 2FA protocol, or a Interfacing Device in front of a Non-GridLogic Field Device
- **Issuer**: Entity that generates and sends commands.
  - May be untrusted (e.g., a lone-wolf insider grid operator or a remote attacker)
- **Validator**: Entity that signs commands and coordinates 2FA
- **Authorizer**: Entity that can approve or deny commands. The second factor in 2FA
  - The authorizer must be trusted, typically a senior engineer or manager
- **GridLogic 2FA Protocol**
  - A protocol that coordinates Field Devices, Issuers, Authorizers, with the Validator and Decision Engine to enable command interception and intervention.



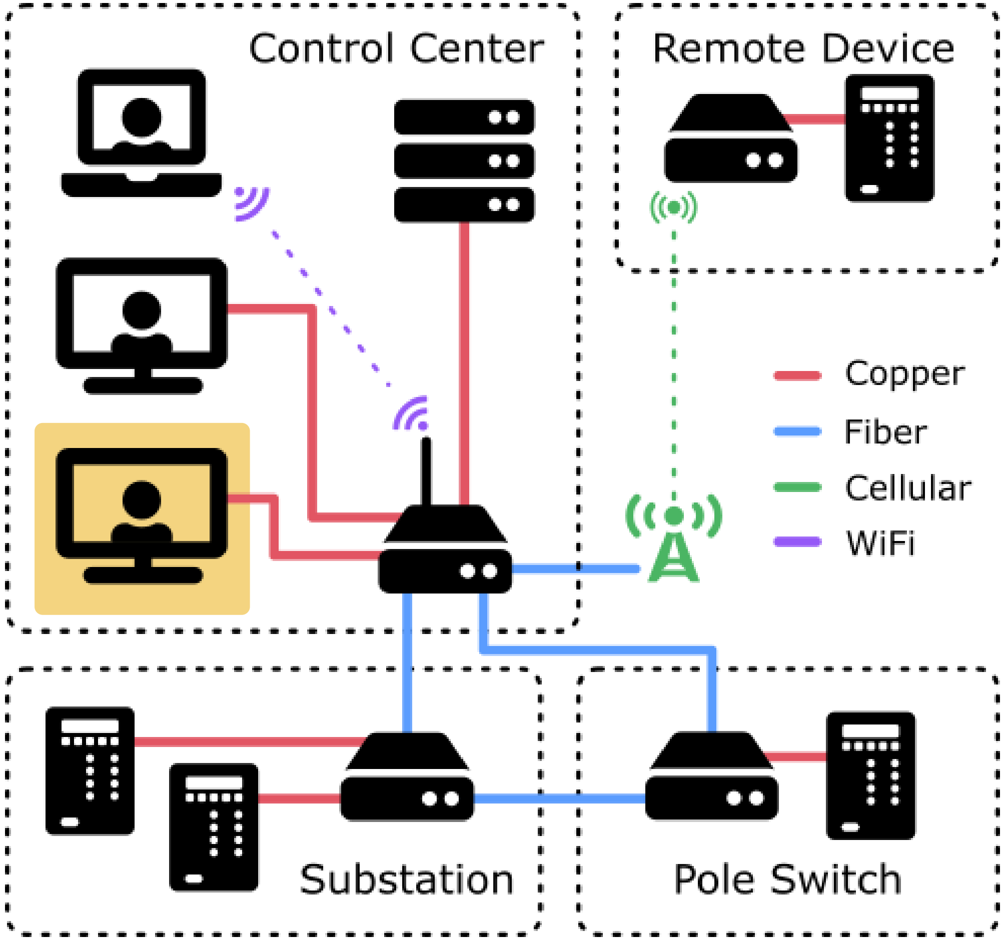
System Architecture

- **Control Center**: Location in a utility where servers, engineers and grid operators are co-located
- **Field Device**: Power-system control device (e.g., digital relay)
- **Interfacing Device**: A device that adds GridLogic 2FA functionality to a Non-GridLogic Field Device
- **GridLogic Device**: A Field Device that supports the GridLogic 2FA protocol, or a Interfacing Device in front of a Non-GridLogic Field Device
- **Issuer**: Entity that generates and sends commands.
  - May be untrusted (e.g., a lone-wolf insider grid operator or a remote attacker)
- **Validator**: Entity that signs commands and coordinates 2FA
- **Authorizer**: Entity that can approve or deny commands. The second factor in 2FA
  - The authorizer must be trusted, typically a senior engineer or manager
- **GridLogic 2FA Protocol**
  - A protocol that coordinates Field Devices, Issuers, Authorizers, with the Validator and Decision Engine to enable command interception and intervention.



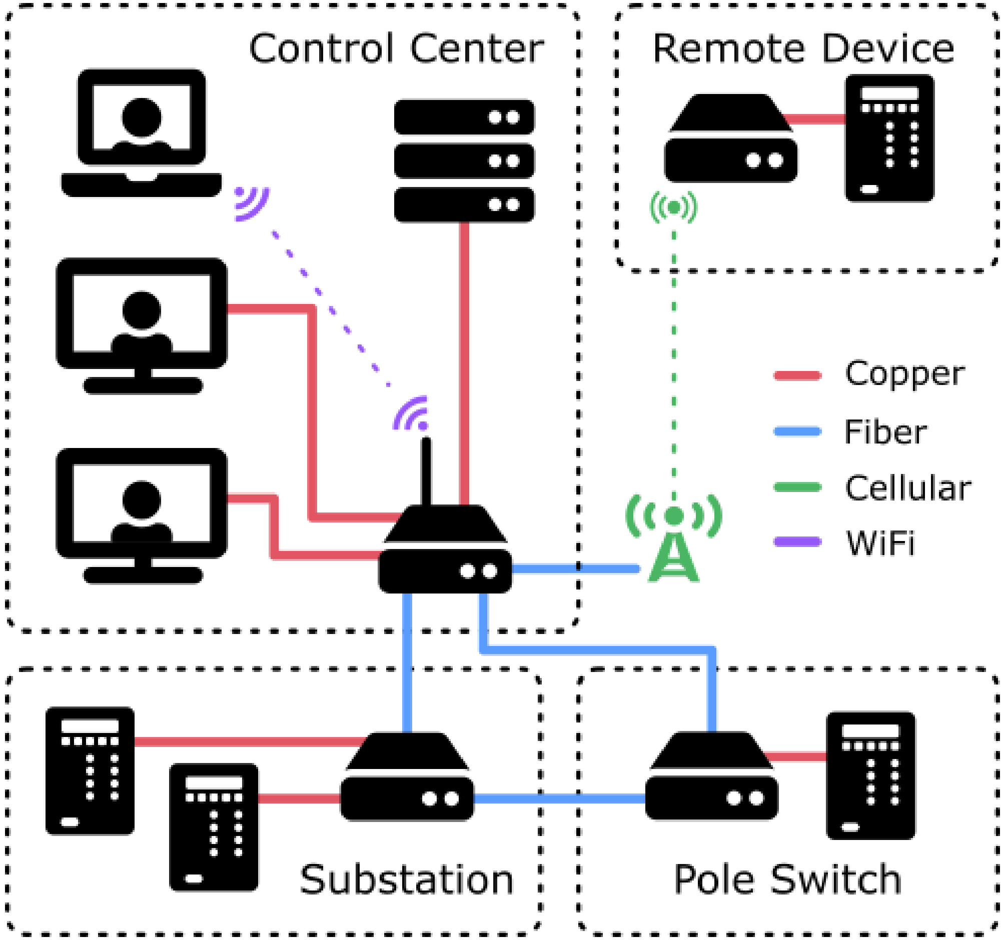
System Architecture

- **Control Center**: Location in a utility where servers, engineers and grid operators are co-located
- **Field Device**: Power-system control device (e.g., digital relay)
- **Interfacing Device**: A device that adds GridLogic 2FA functionality to a Non-GridLogic Field Device
- **GridLogic Device**: A Field Device that supports the GridLogic 2FA protocol, or a Interfacing Device in front of a Non-GridLogic Field Device
- **Issuer**: Entity that generates and sends commands.
  - May be untrusted (e.g., a lone-wolf insider grid operator or a remote attacker)
- **Validator**: Entity that signs commands and coordinates 2FA
- **Authorizer**: Entity that can approve or deny commands. The second factor in 2FA
  - The authorizer must be trusted, typically a senior engineer or manager
- **GridLogic 2FA Protocol**
  - A protocol that coordinates Field Devices, Issuers, Authorizers, with the Validator and Decision Engine to enable command interception and intervention.



System Architecture

- **Control Center**: Location in a utility where servers, engineers and grid operators are co-located
- **Field Device**: Power-system control device (e.g., digital relay)
- **Interfacing Device**: A device that adds GridLogic 2FA functionality to a Non-GridLogic Field Device
- **GridLogic Device**: A Field Device that supports the GridLogic 2FA protocol, or a Interfacing Device in front of a Non-GridLogic Field Device
- **Issuer**: Entity that generates and sends commands.
  - May be untrusted (e.g., a lone-wolf insider grid operator or a remote attacker)
- **Validator**: Entity that signs commands and coordinates 2FA
- **Authorizer**: Entity that can approve or deny commands. The second factor in 2FA
  - The authorizer must be trusted, typically a senior engineer or manager
- **GridLogic 2FA Protocol**
  - A protocol that coordinates Field Devices, Issuers, Authorizers, with the Validator and Decision Engine to enable command interception and intervention.

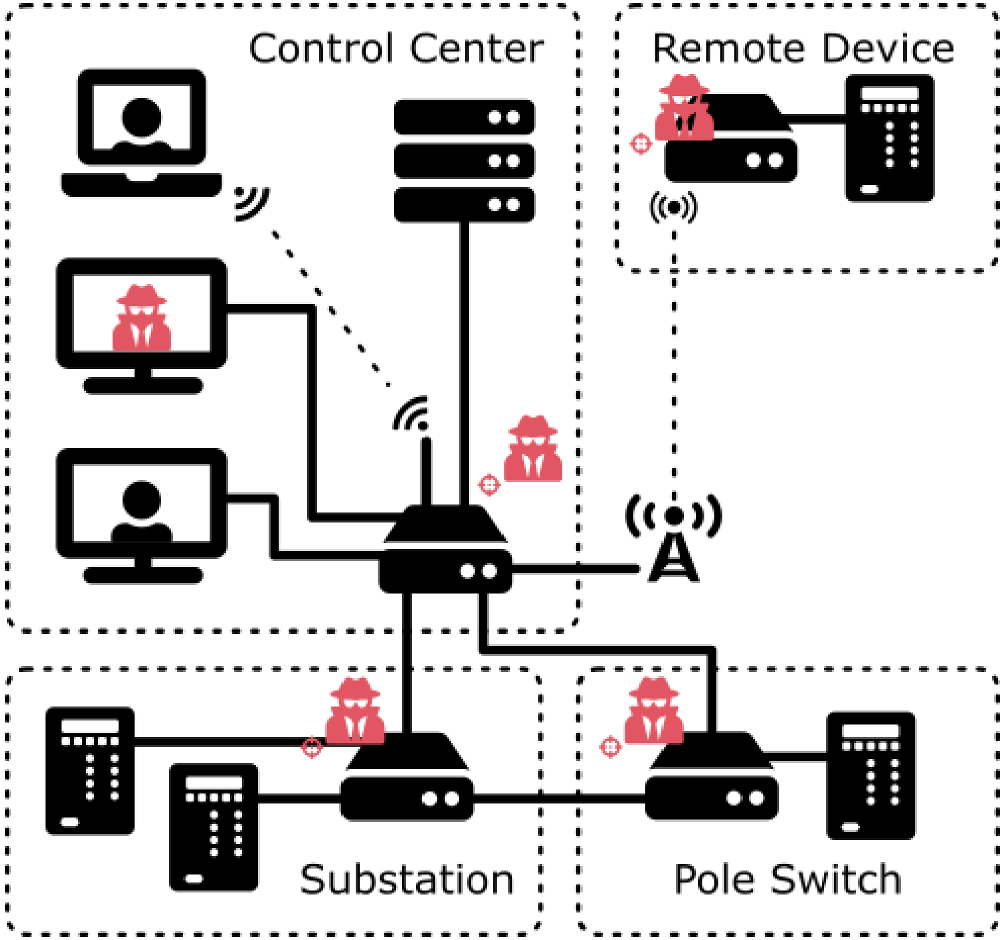


Outline

1. Problem Statement
2. System Architecture
3. **Threat Model**
4. 2FA Protocol
5. Experiments
    1. Security Experiments
    2. Performance Experiments
6. Discussion
7. References

Threat Model

- Issuers may be untrusted
  - **Lone Wolf Insider**: Low-Level Engineer sending malicious commands, with access to their own low-level signing key
  - **Remote Attacker**: Attacker connects to SCADA network and sends malicious commands without access to a signing key
- Issuers do not have root access to Field Devices
- Authorizers are always trusted

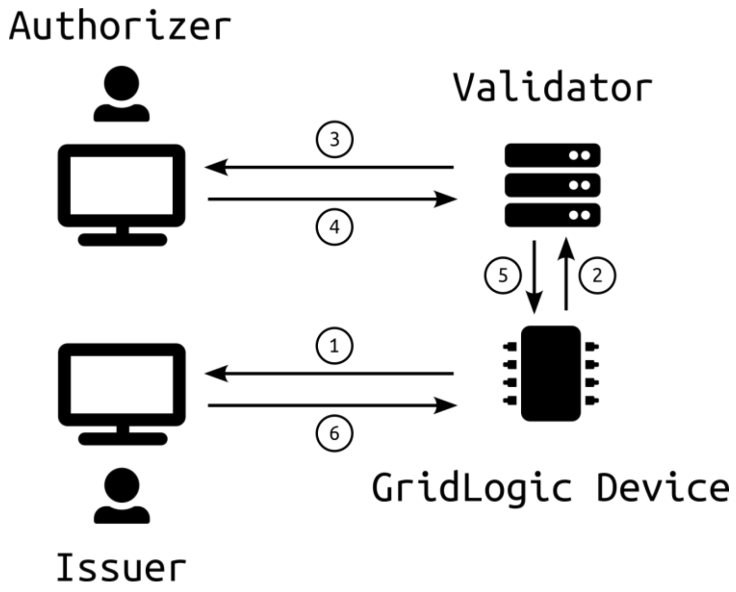


Outline

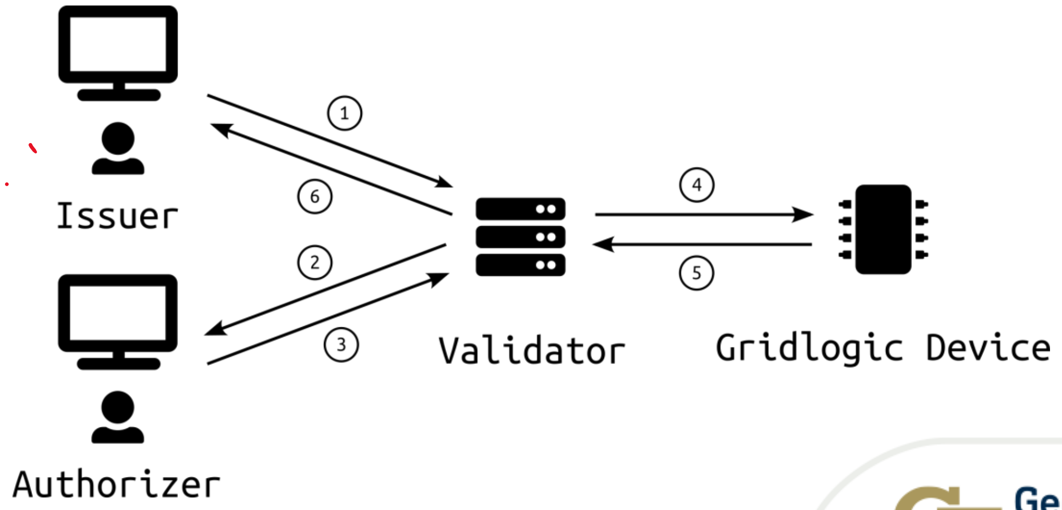
1. Problem Statement
2. System Architecture
3. Threat Model
4. **2FA Protocol**
5. Experiments
    1. Security Experiments
    2. Performance Experiments
6. Discussion
7. References

GridLogic 2FA Protocol

- Commands are intercepted before being executed and analyzed for sensitivity
  - Commands without a valid signature are dropped entirely
- Sensitive Commands are sent to an Authorizer for a second factor of approval
- In Edge Intercept Mode (EIM) commands are intercepted at the device
- In Control Center Intercept Mode (CCIM) commands are intercepted at the validator in the control center



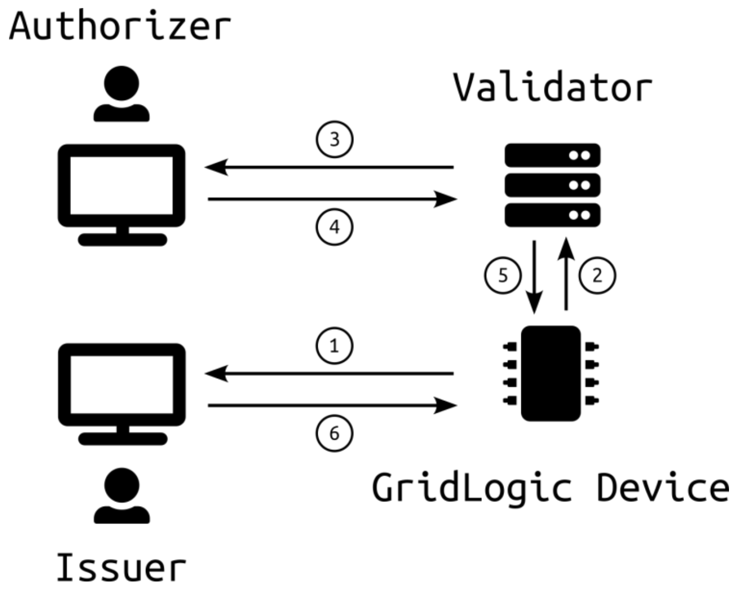
EIM



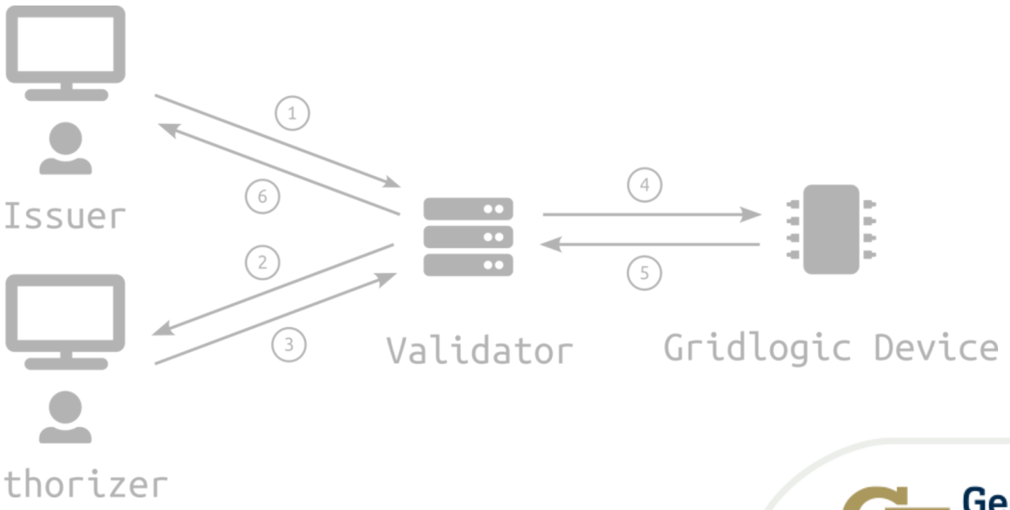
CCIM

GridLogic 2FA Protocol

- Commands are intercepted before being executed and analyzed for sensitivity
  - Commands without a valid signature are dropped entirely
- Sensitive Commands are sent to an Authorizer for a second factor of approval
- In Edge Intercept Mode (EIM) commands are intercepted at the device
- In Control Center Intercept Mode (CCIM) commands are intercepted at the validator in the control center



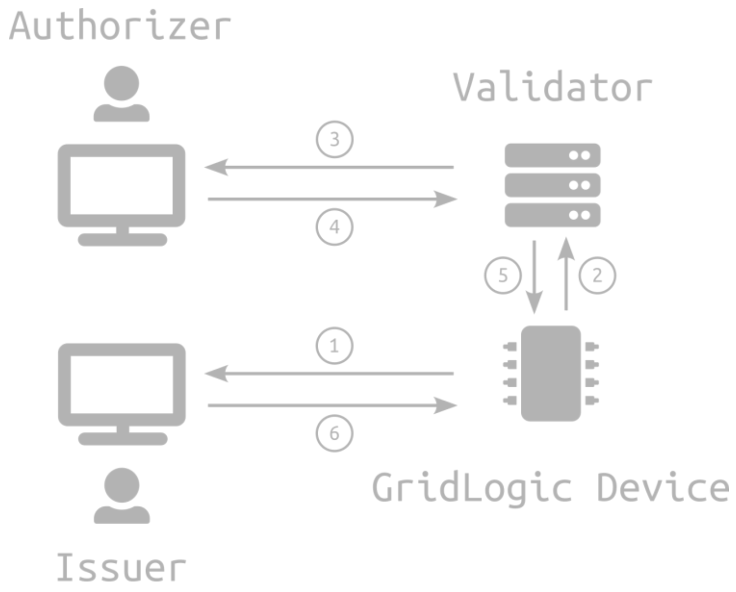
EIM



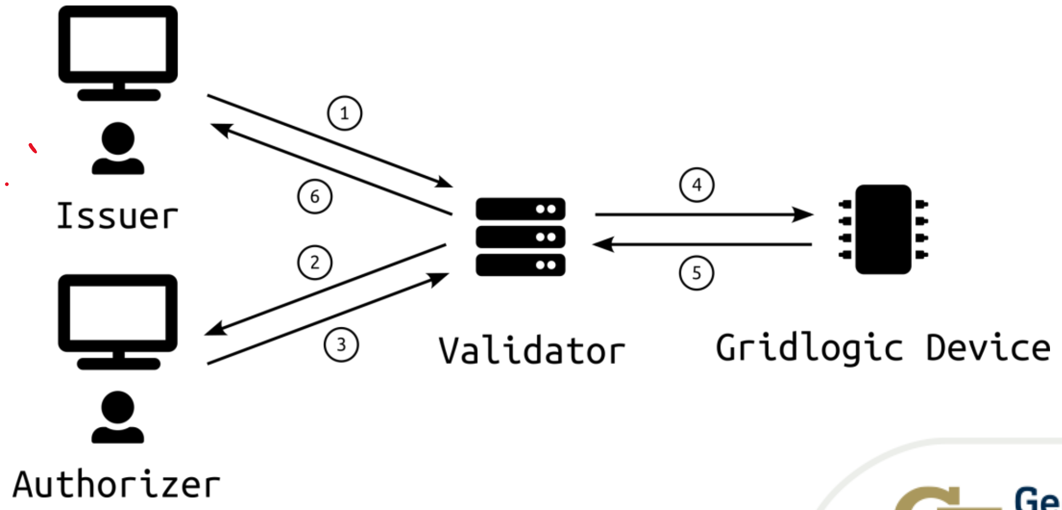
CCIM

GridLogic 2FA Protocol

- Commands are intercepted before being executed and analyzed for sensitivity
    - Commands without a valid signature are dropped entirely
- Sensitive Commands are sent to an Authorizer for a second factor of approval
- In Edge Intercept Mode (EIM) commands are intercepted at the device
- In Control Center Intercept Mode (CCIM) commands are intercepted at the validator in the control center



EIM



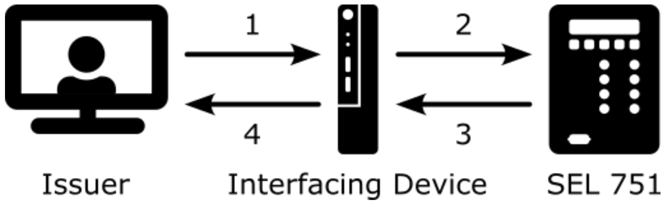
CCIM

Outline

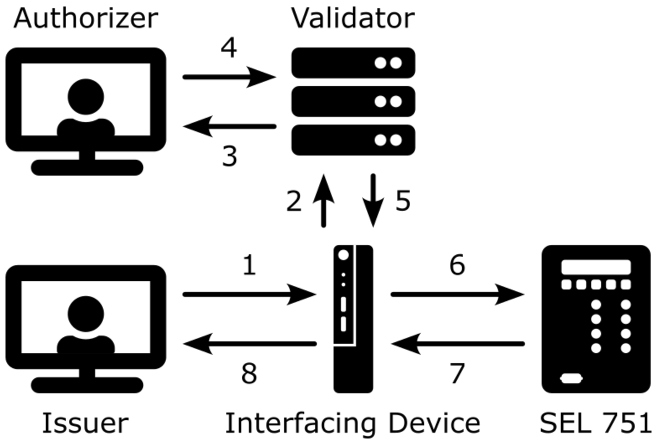
1. Problem Statement
2. System Architecture
3. Threat Model
4. 2FA Protocol
5. **Experiments**
    1. Security Experiments
    2. Performance Experiments
6. Discussion
7. References

Experimental Setup

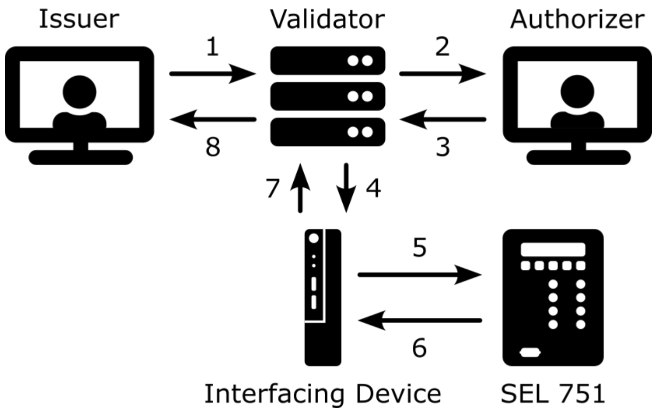
- **Interfacing Device**: HTTP Server on Dell Mini-PC
- **Field Device**: SEL 751 (serial protocol)
- **Issuer**: Python CLI on Dell Laptop
- **Authorizer**: Firefox browser on Dell Laptop
- **Validator**: HTTP Server on MacBook



Base Case



EIM

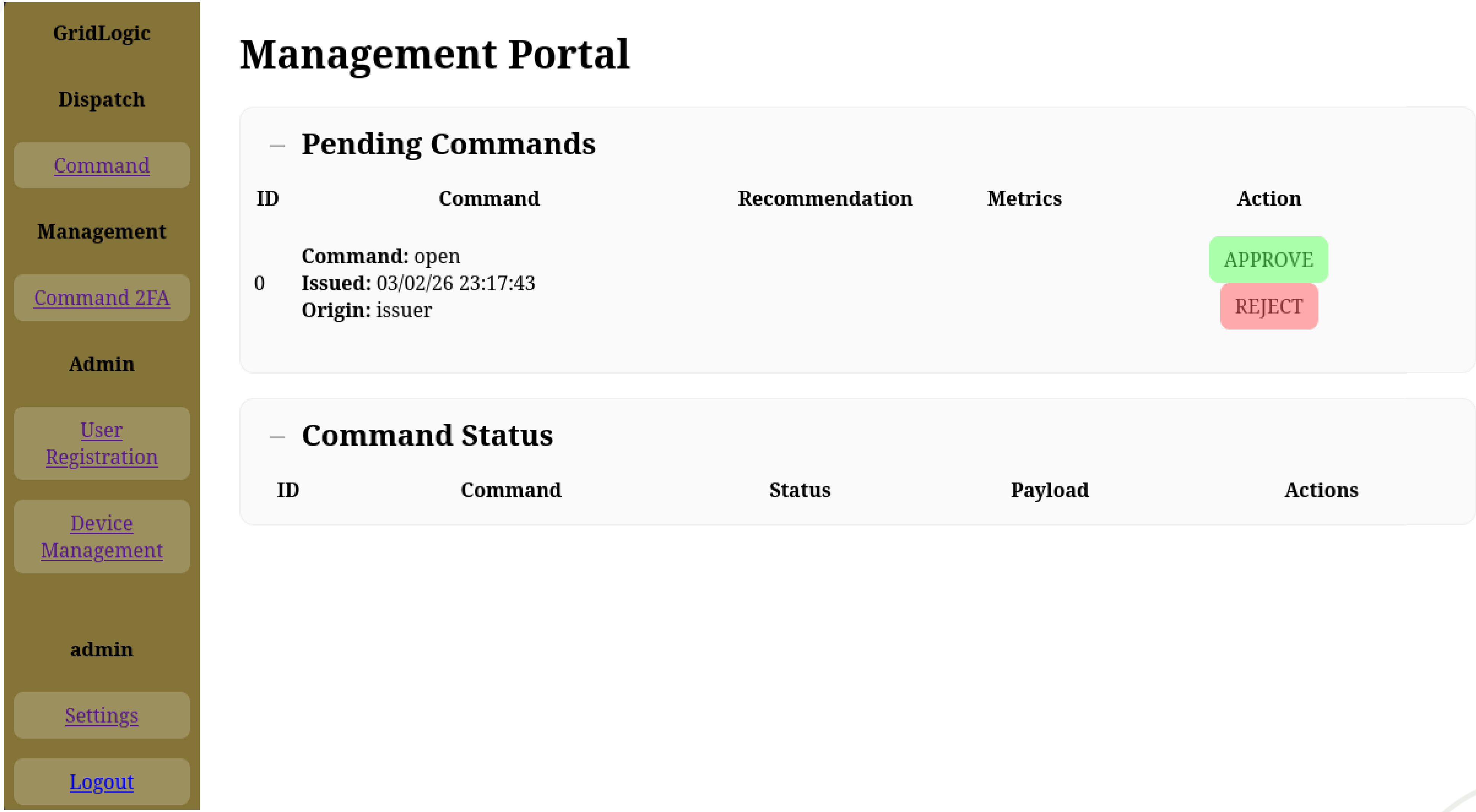


CCIM

Experimental Setup: Interfacing Device

- We select the following three commands for our experiments
  - Open (trip relay)
  - Read (read relay status)
  - Reset (clear tripped status)

Experimental Setup: Authorizer Interface



Management Portal

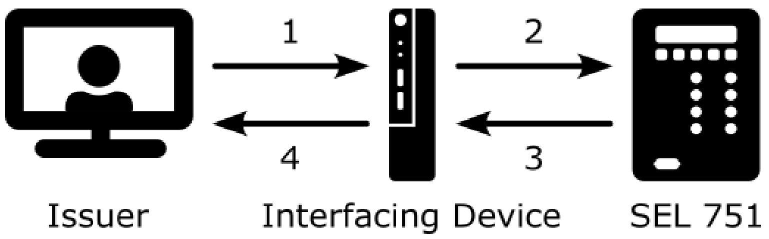
– Pending Commands

ID	Command	Recommendation	Metrics	Action
0	**Command:** open **Issued:** 03/02/26 23:17:43 **Origin:** issuer			APPROVE REJECT

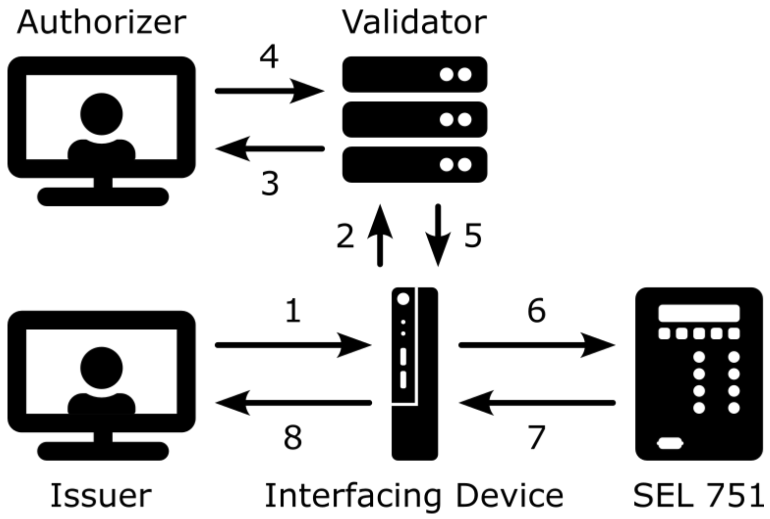
– Command Status

ID	Command	Status	Payload	Actions

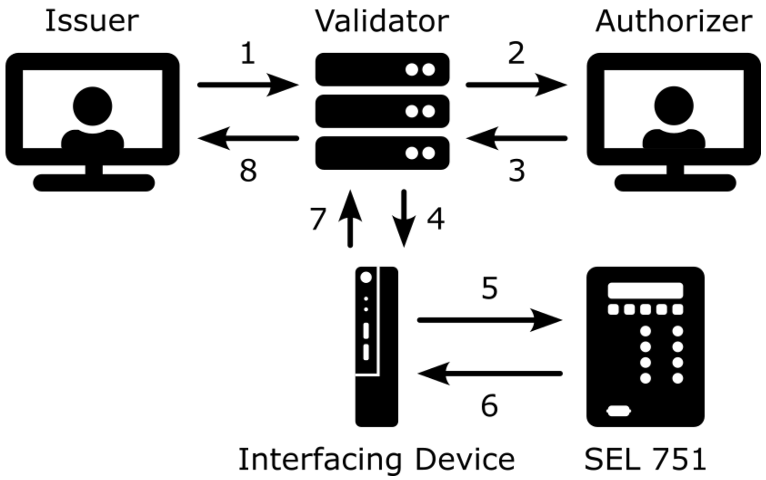
Experimental Setup: Modes of Operation



Base Case

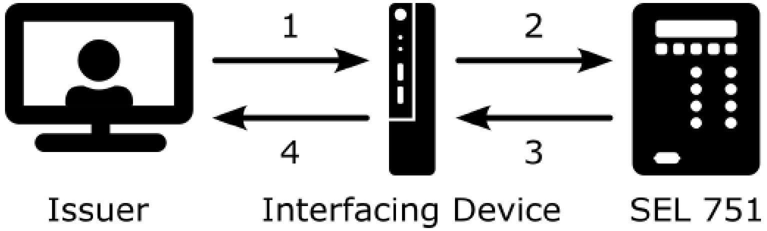


EIM

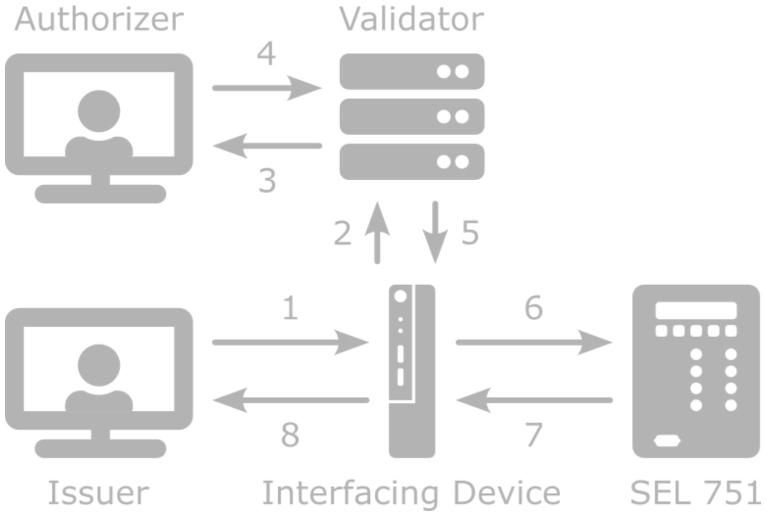


CCIM

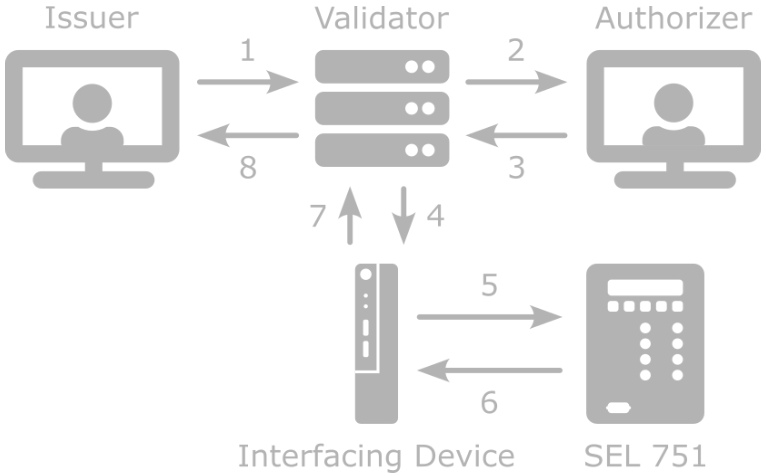
Experimental Setup: Modes of Operation



Base Case

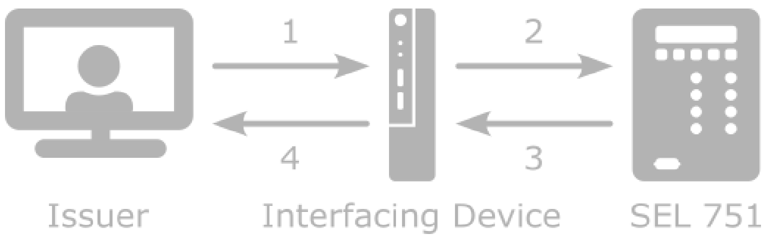


EIM

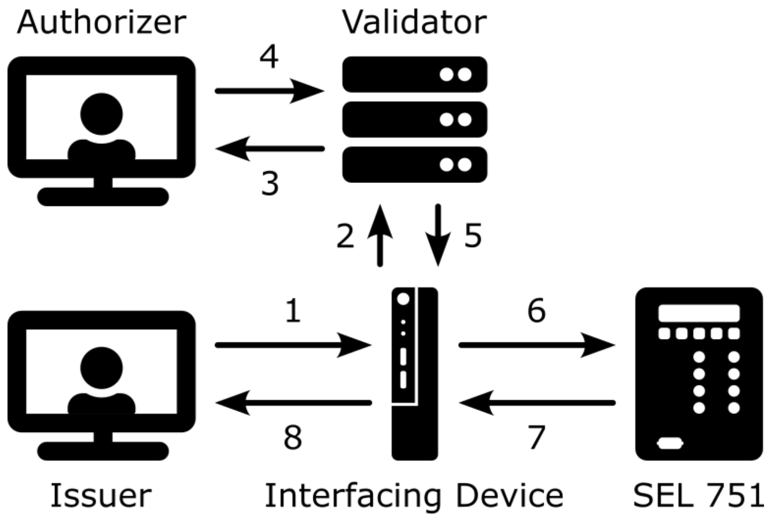


CCIM

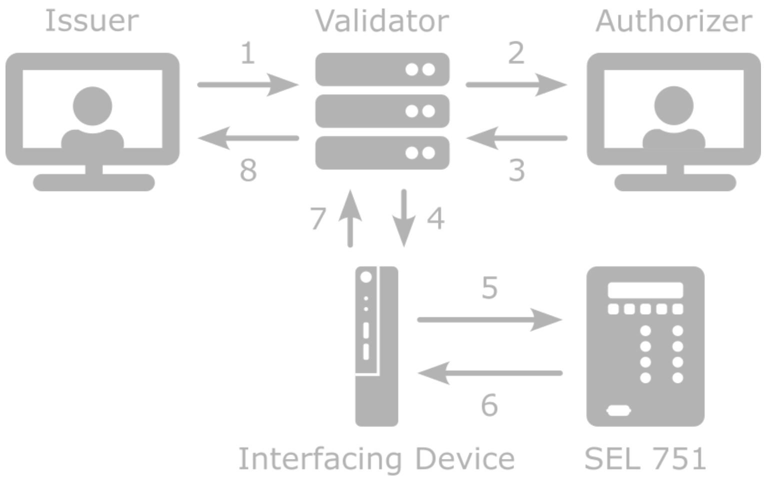
Experimental Setup: Modes of Operation



Base Case

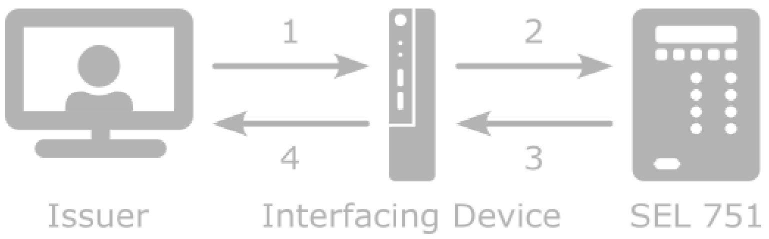


EIM

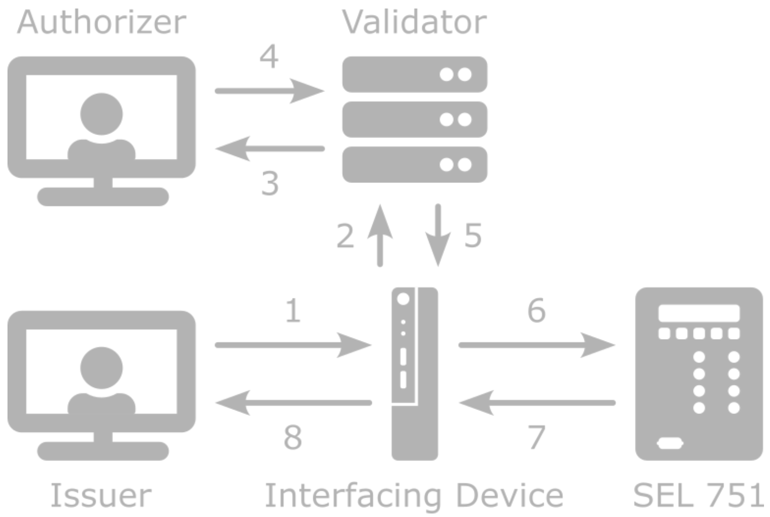


CCIM

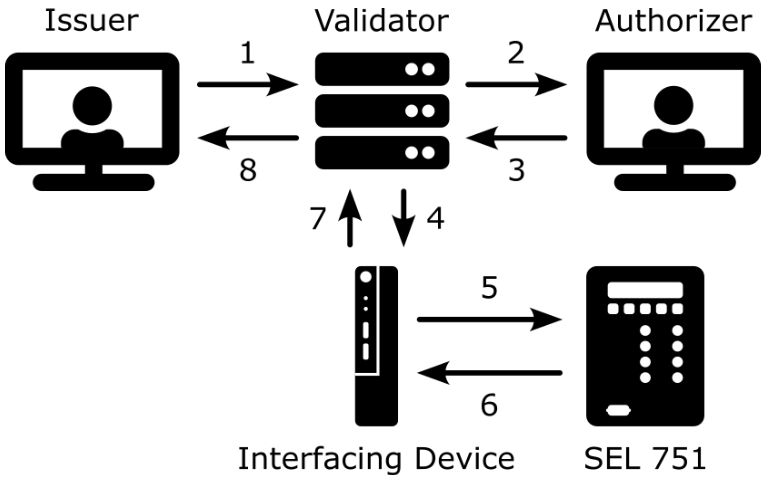
Experimental Setup: Modes of Operation



Base Case



EIM



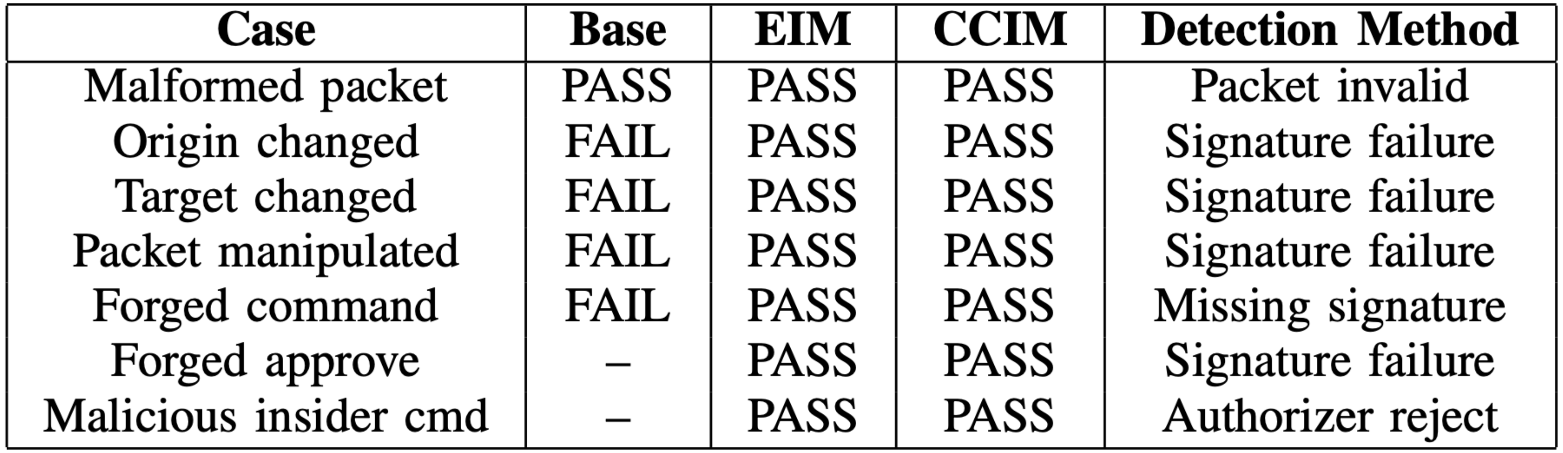
CCIM

Outline

1. Problem Statement
2. System Architecture
3. Threat Model
4. 2FA Protocol
5. Experiments
    1. **Security Experiments**
    2. Performance Experiments
6. Discussion
7. References

Security Experiments

TABLE I: Security Tests



Case	Base	EIM	CCIM	Detection Method
Malformed packet	PASS	PASS	PASS	Packet invalid
Origin changed	FAIL	PASS	PASS	Signature failure
Target changed	FAIL	PASS	PASS	Signature failure
Packet manipulated	FAIL	PASS	PASS	Signature failure
Forged command	FAIL	PASS	PASS	Missing signature
Forged approve	–	PASS	PASS	Signature failure
Malicious insider cmd	–	PASS	PASS	Authorizer reject

PASS: Modification detected and packet dropped
FAIL: Modification not detected and packet allowed to proceed

Outline

1. Problem Statement
2. System Architecture
3. Threat Model
4. 2FA Protocol
5. Experiments
    1. Security Experiments
    2. **Performance Experiments**
6. Discussion
7. References

Performance Experiments: SSL Disabled

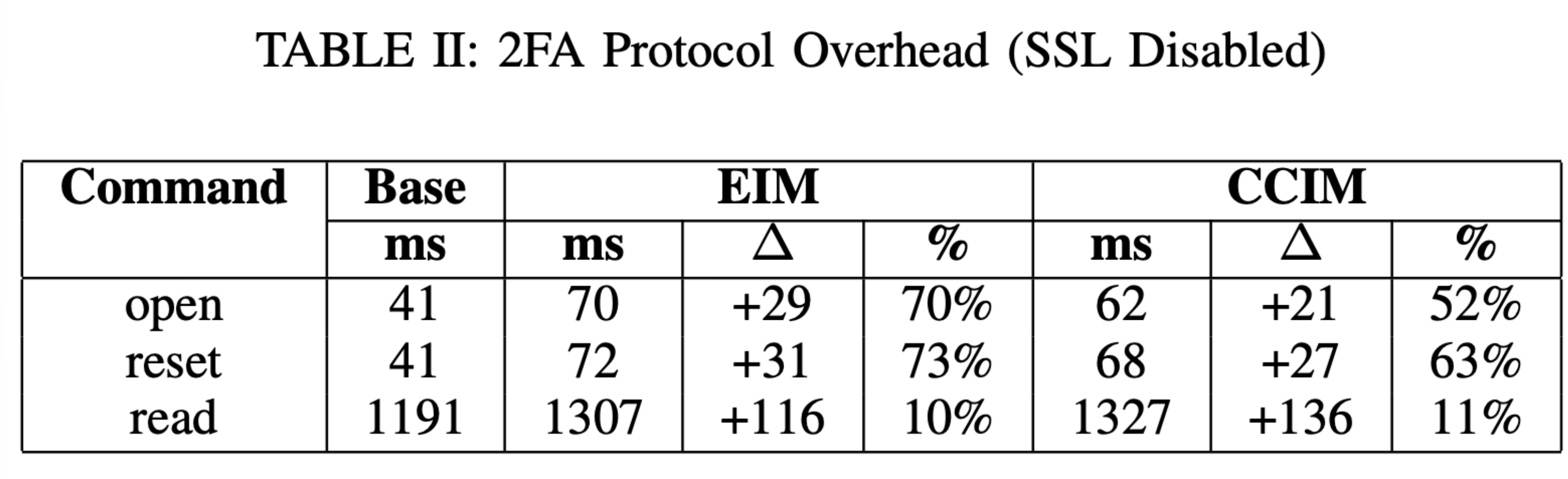


TABLE II: 2FA Protocol Overhead (SSL Disabled)

Command	**Base**	**EIM**			**CCIM**		
	ms	**ms**	**Δ**	**%**	**ms**	**Δ**	**%**
open	41	70	+29	70%	62	+21	52%
reset	41	72	+31	73%	68	+27	63%
read	1191	1307	+116	10%	1327	+136	11%

Performance Experiments: SSL Disabled

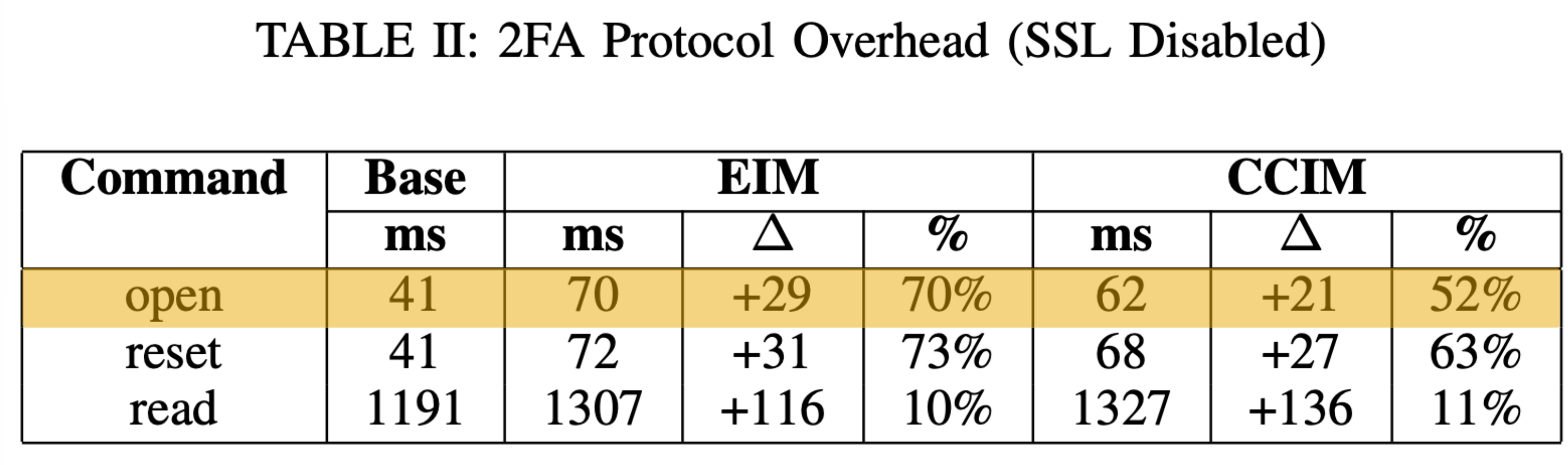


TABLE II: 2FA Protocol Overhead (SSL Disabled)

Command	Base	EIM			CCIM		
	ms	ms	Δ	%	ms	Δ	%
open	41	70	+29	70%	62	+21	52%
reset	41	72	+31	73%	68	+27	63%
read	1191	1307	+116	10%	1327	+136	11%

Performance Experiments: SSL Disabled

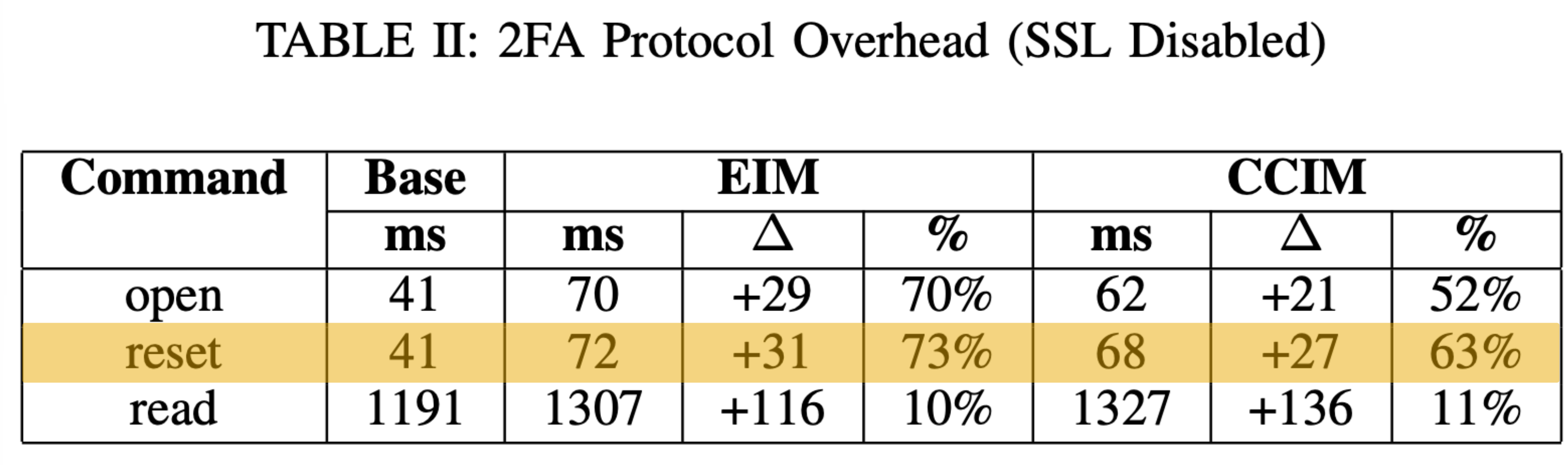


TABLE II: 2FA Protocol Overhead (SSL Disabled)

Command	Base	EIM			CCIM		
	ms	ms	Δ	%	ms	Δ	%
open	41	70	+29	70%	62	+21	52%
reset	41	72	+31	73%	68	+27	63%
read	1191	1307	+116	10%	1327	+136	11%

Performance Experiments: SSL Disabled

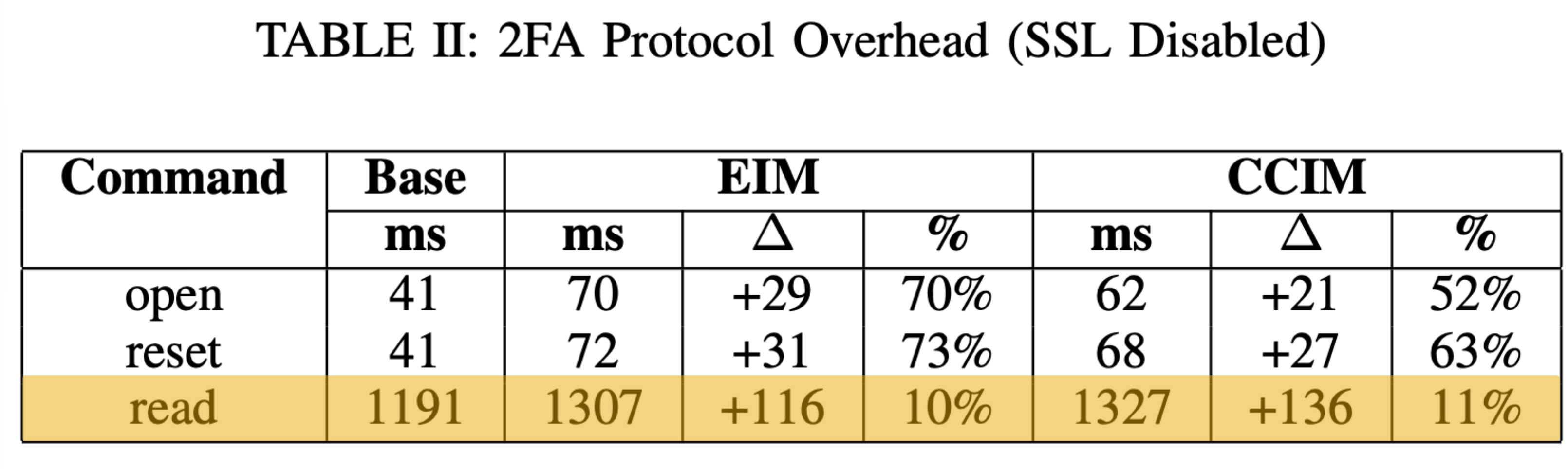


TABLE II: 2FA Protocol Overhead (SSL Disabled)

Command	Base	EIM			CCIM		
	ms	ms	Δ	%	ms	Δ	%
open	41	70	+29	70%	62	+21	52%
reset	41	72	+31	73%	68	+27	63%
read	1191	1307	+116	10%	1327	+136	11%

Performance Experiments: SSL Disabled

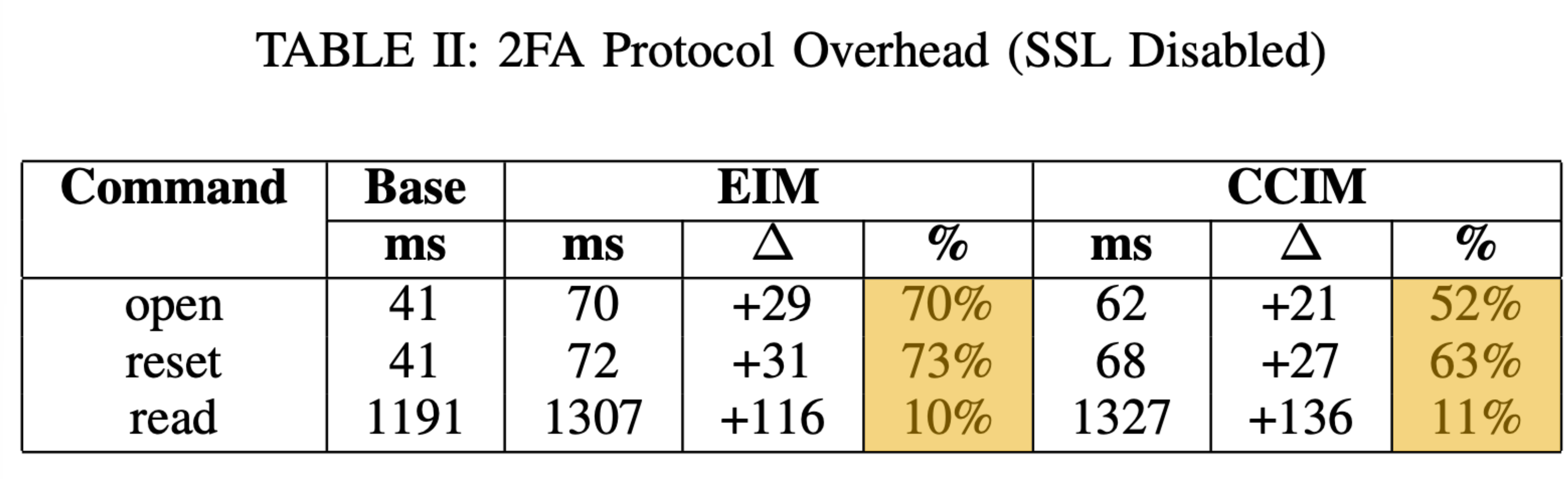


TABLE II: 2FA Protocol Overhead (SSL Disabled)

Command	Base	EIM			CCIM		
	ms	ms	Δ	%	ms	Δ	%
open	41	70	+29	70%	62	+21	52%
reset	41	72	+31	73%	68	+27	63%
read	1191	1307	+116	10%	1327	+136	11%

Performance Experiments: SSL Enabled

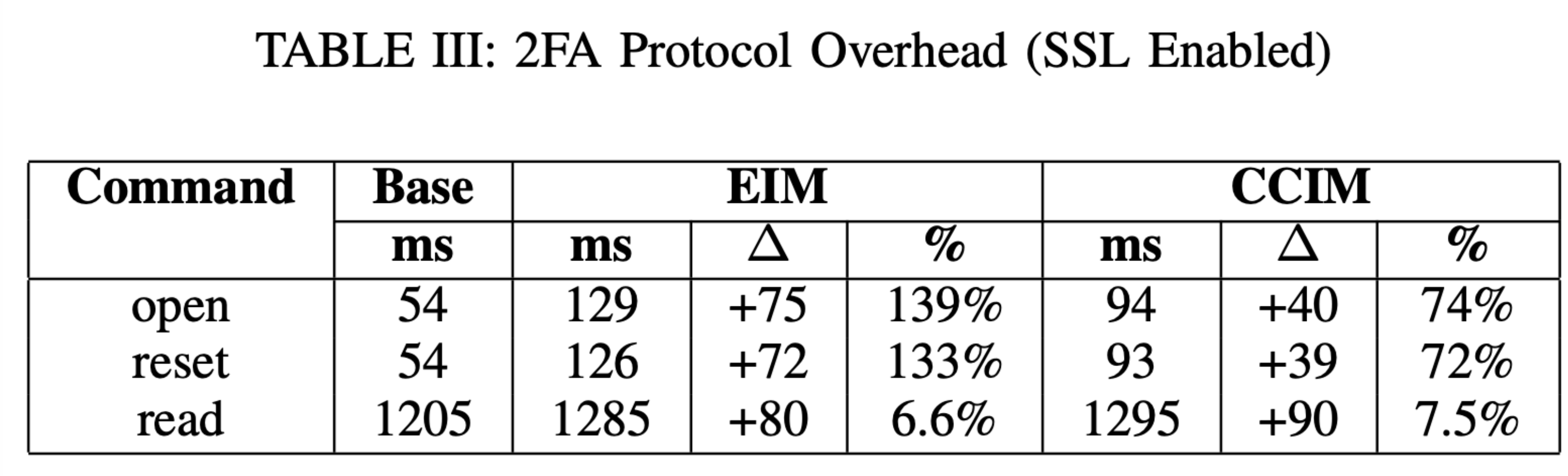


TABLE III: 2FA Protocol Overhead (SSL Enabled)

Command	Base	EIM			CCIM		
	ms	ms	Δ	%	ms	Δ	%
open	54	129	+75	139%	94	+40	74%
reset	54	126	+72	133%	93	+39	72%
read	1205	1285	+80	6.6%	1295	+90	7.5%

Performance Experiments: SSL Enabled

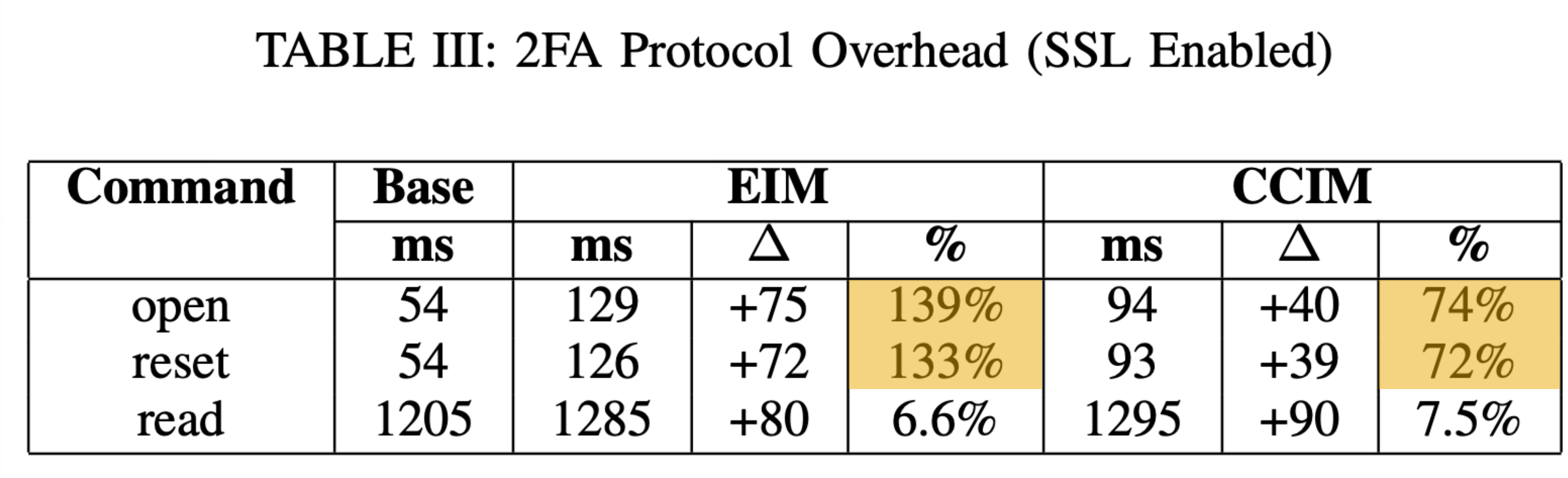


TABLE III: 2FA Protocol Overhead (SSL Enabled)

Command	Base	EIM			CCIM		
	ms	ms	Δ	%	ms	Δ	%
open	54	129	+75	139%	94	+40	74%
reset	54	126	+72	133%	93	+39	72%
read	1205	1285	+80	6.6%	1295	+90	7.5%

Performance Experiments: SSL Enabled

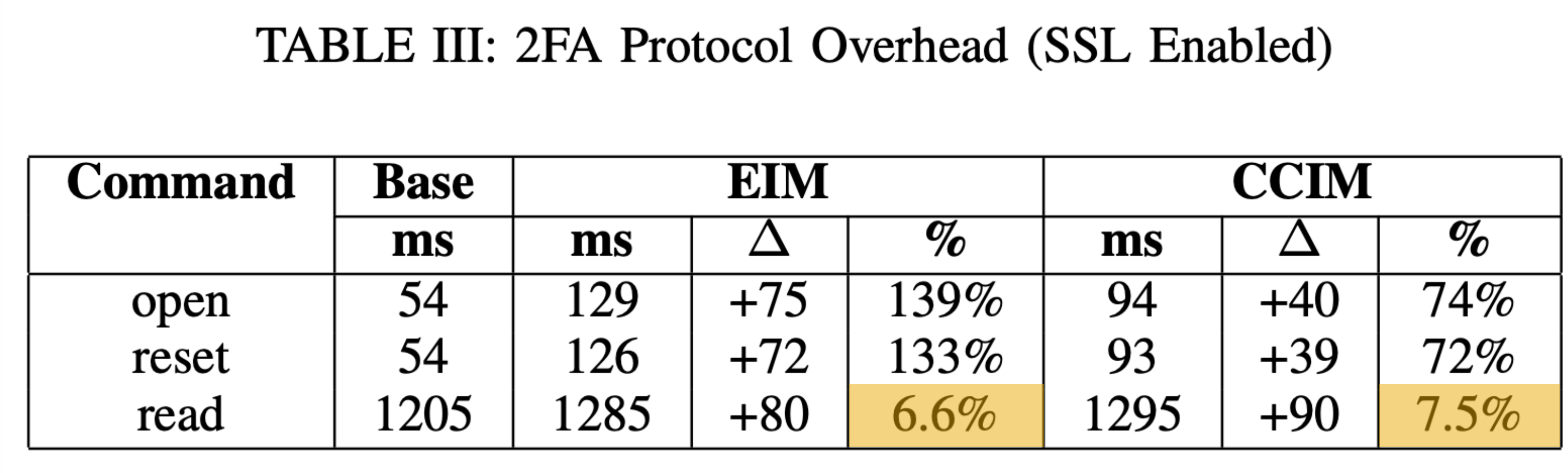


TABLE III: 2FA Protocol Overhead (SSL Enabled)

Command	Base	EIM			CCIM		
	ms	ms	Δ	%	ms	Δ	%
open	54	129	+75	139%	94	+40	74%
reset	54	126	+72	133%	93	+39	72%
read	1205	1285	+80	6.6%	1295	+90	7.5%

Performance Experiments: SSL Overhead

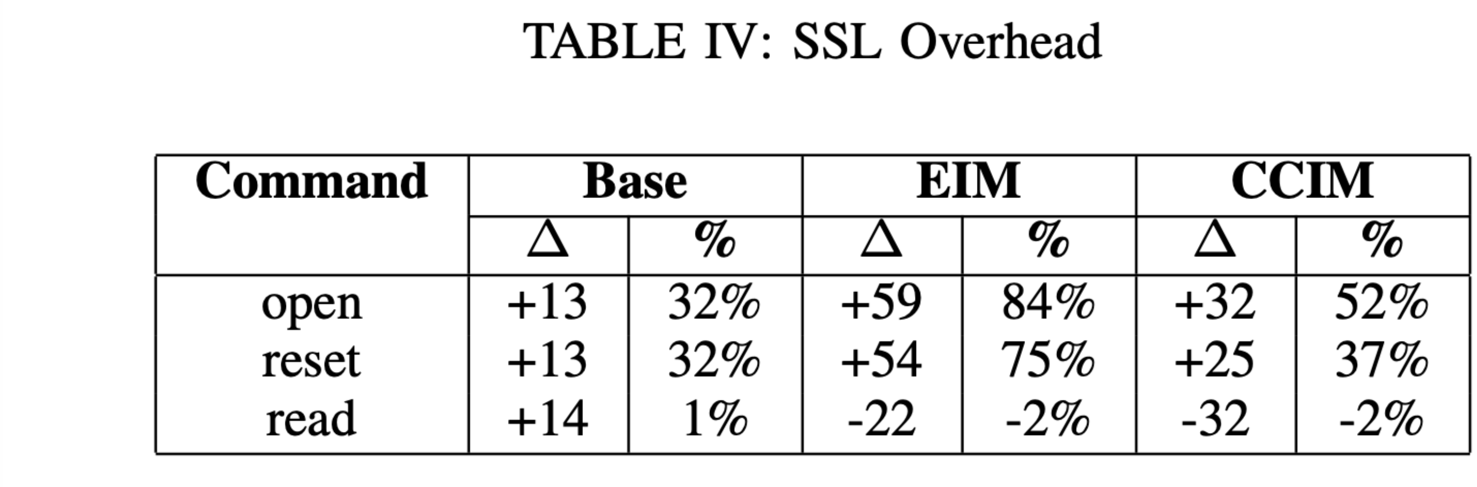


TABLE IV: SSL Overhead

Command	**Base**		**EIM**		**CCIM**	
	Δ	%	Δ	%	Δ	%
open	+13	32%	+59	84%	+32	52%
reset	+13	32%	+54	75%	+25	37%
read	+14	1%	-22	-2%	-32	-2%

Performance Experiments: SSL Overhead

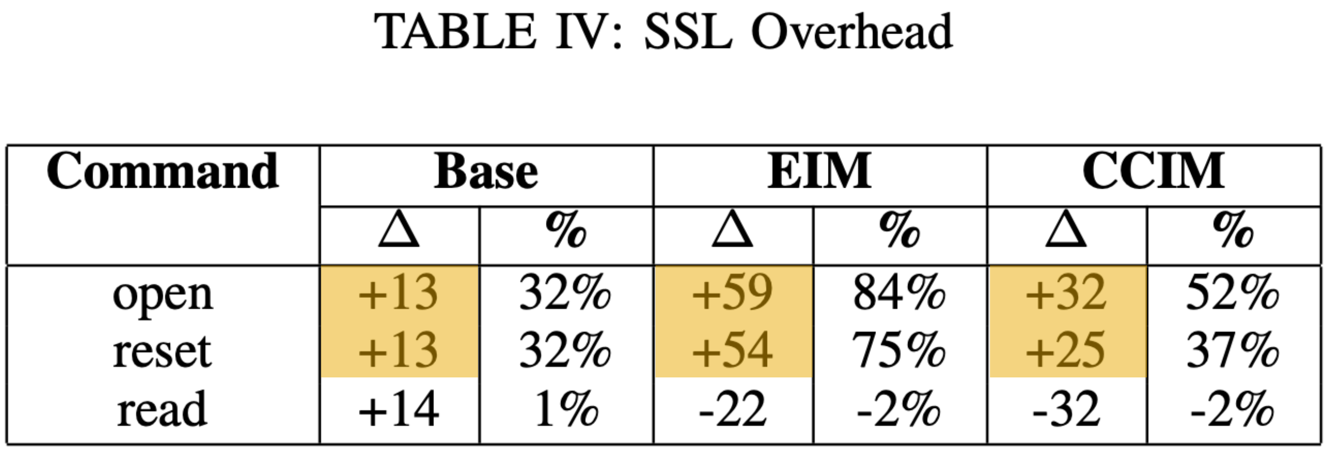


TABLE IV: SSL Overhead

Command	Base		EIM		CCIM	
	Δ	%	Δ	%	Δ	%
open	+13	32%	+59	84%	+32	52%
reset	+13	32%	+54	75%	+25	37%
read	+14	1%	-22	-2%	-32	-2%

Performance Experiments: SSL Overhead

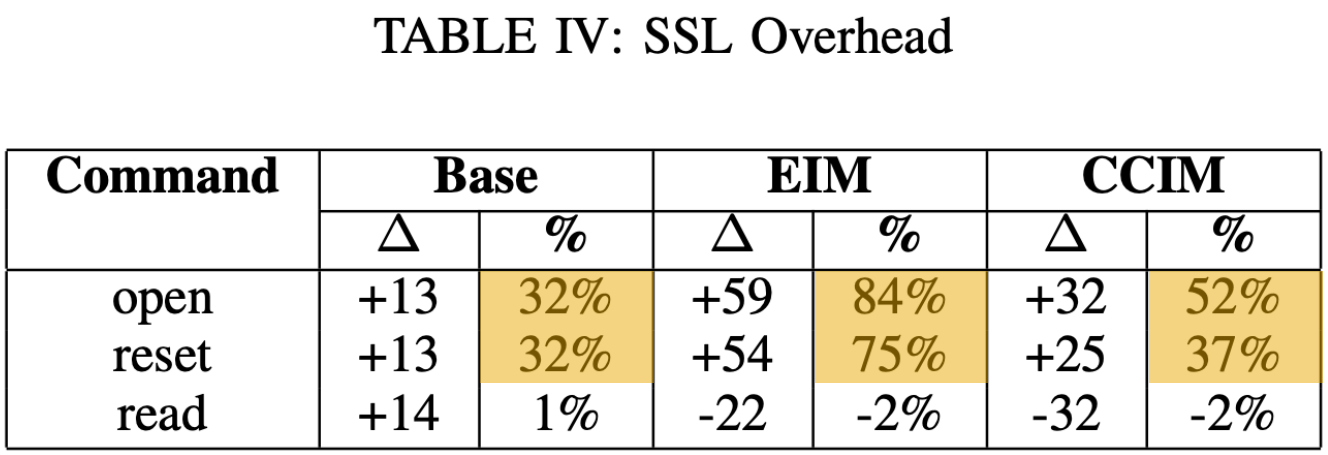


TABLE IV: SSL Overhead

Command	Base		EIM		CCIM	
	Δ	%	Δ	%	Δ	%
open	+13	32%	+59	84%	+32	52%
reset	+13	32%	+54	75%	+25	37%
read	+14	1%	-22	-2%	-32	-2%

Outline

1. Problem Statement
2. System Architecture
3. Threat Model
4. 2FA Protocol
5. Experiments
    1. Security Experiments
    2. Performance Experiments
6. **Discussion**
7. References

Discussion

- Protocol Overheads:
  - Worst observed protocol overhead: 136 ms
  - Average reaction time based on lexical comprehension of a word: 871 ms [15]
  - Protocol overhead dwarfed by the human in 2FA: 136 / 871 = 15.6%
- SSL Overhead:
  - Worst observed overhead: 59ms, 84%
- Scalability:
  - Utilities need to consider their own specific timing requirements and network architecture since topology and round-trip times vary widely depending on coverage and geography

Outline

1. Problem Statement
2. System Architecture
3. Threat Model
4. 2FA Protocol
5. Experiments
    1. Security Experiments
    2. Performance Experiments
6. Discussion
7. **References**

References

[1] B. E. Humphreys, “Attacks on Ukraine’s Electric Grid: Insights for U.S. Infrastructure Security and Resilience,” May 2024. [Online]. Available: https://www.congress.gov/crs-product/R48067

[2] Check Point Research, “The State of Cyber Security 2025: Top threats, emerging trends, and CISO recommendations,” September 2025. [Online]. Available: https://www.checkpoint.com/security-report/

[3] D. o. E. Office of Electricity, “Secure Communications Interoperability in the Power Grid,” September 2023. [Online]. Available: https://www.energy.gov/oe/grid-communications-and-security

[4] Schweitzer Engineering Laboratories Inc., “SEL-751 Feeder Protection Relay,” https://selinc.com/products/751/, 2025.

[5] M. Masse, REST API Design Rulebook. O’Reilly, 2011.

[6] WHATWG, “HTML Living Standard,” Web Hypertext Application Technology Working Group, Tech. Rep., 2025. [Online]. Available: https://html.spec.whatwg.org/multipage/

[7] H. Krawczyk, M. Bellare, and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” Interet Requests for Comments, RFC Editor, RFC, February 1997. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2104

[8] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, HANDBOOK of APPLIED CRYPTOGRAPHY. CRC Press, 1996, p. 390.

[9] Pallets, “Flask,” https://flask.palletsprojects.com/en/stable/, 2025.

[10] uWSGI, “uWSGI,” https://uwsgi-docs.readthedocs.io/en/latest/, 2025.

[11] Igor Sysoev and F5 Inc, “nginx,” https://nginx.org/, 2025.

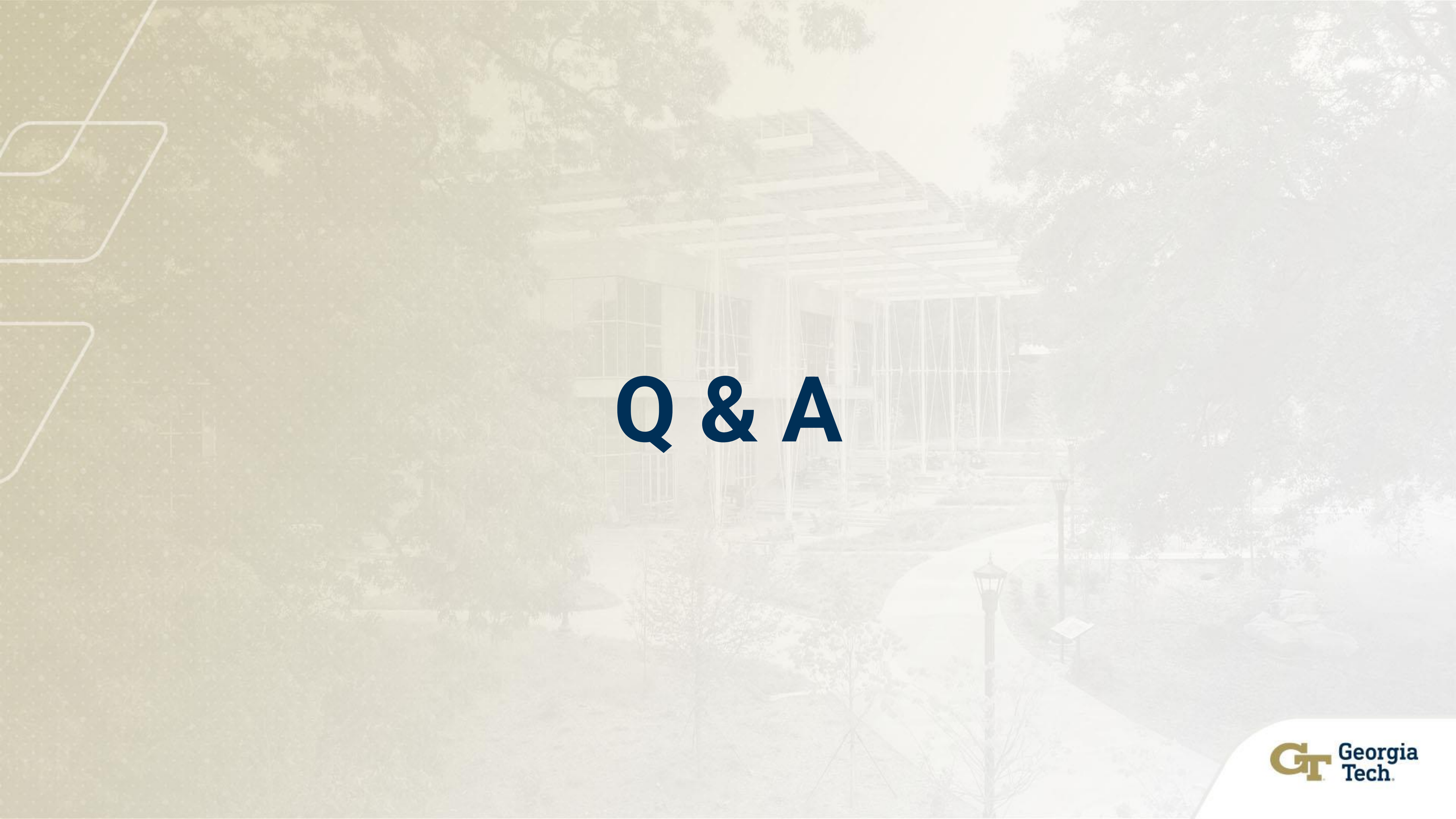
[12] C. PERCIVAL, “Stronger key derivation via sequential memory-hard functions,” tarsnap, Tech. Rep., 2009. [Online]. Available: https://www.tarsnap.com/scrypt/scrypt.pdf

[13] C. Percival and S. Josefsson, “The scrypt password-based key derivation function,” Interet Requests for Comments, RFC Editor, RFC, August 2016. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7914

[14] R. McGill, J. W. Tukey, and W. A. Larsen, “Variations of box plots,” The American Statistician, vol. 32, no. 1, pp. 12–16, 1978. [Online]. Available: http://www.jstor.org/stable/2683468

[15] O. Hauk, C. Coutout, A. Holden, and Y. Chen, “The time-course of single-word reading: Evidence from fast behavioral and brain responses,” NeuroImage, vol. 60, no. 2, pp. 1462–1477, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S105381191200078X

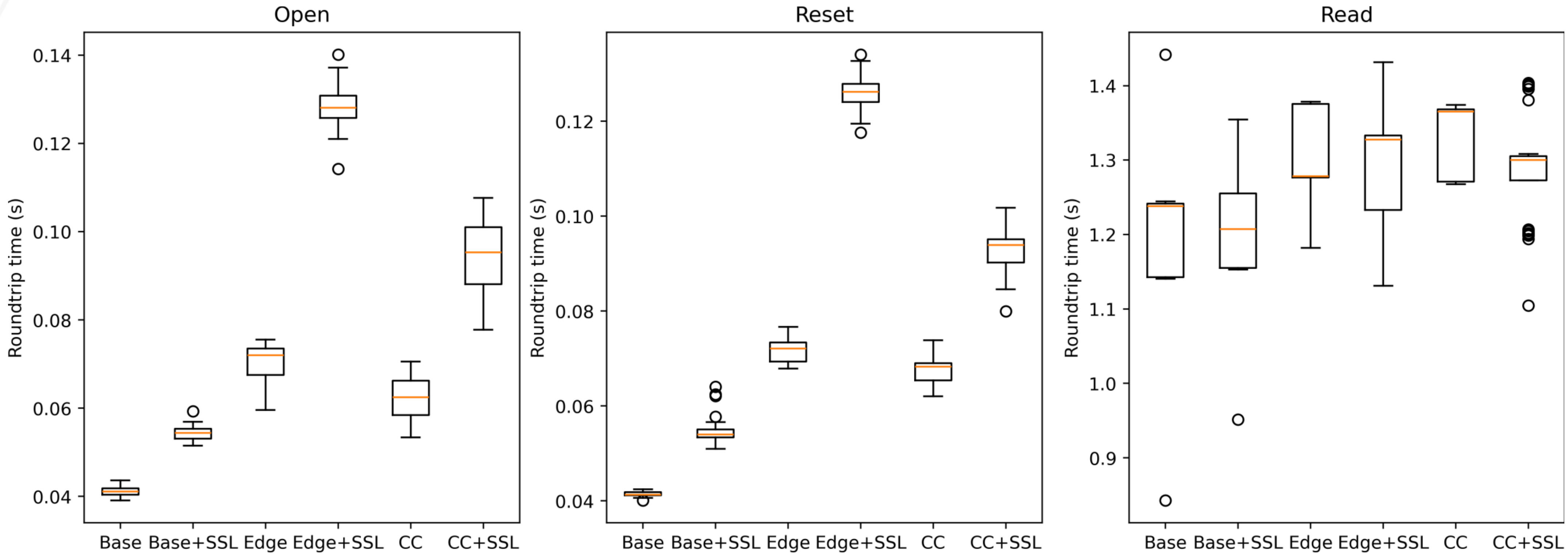
[16] P. K. Wan, A. Satybaldy, L. Huang, H. Holtskog, and M. Nowostawski, “Reducing alert fatigue by sharing low-level alerts with patients and enhancing collaborative decision making using blockchain technology: Scoping review and proposed framework (medalert),” J Med Internet Res, vol. 22, no. 10, p. e22013, Oct 2020. [Online]. Available: http://www.jmir.org/2020/10/e22013/



Q & A

Appendix

Performance Experiments



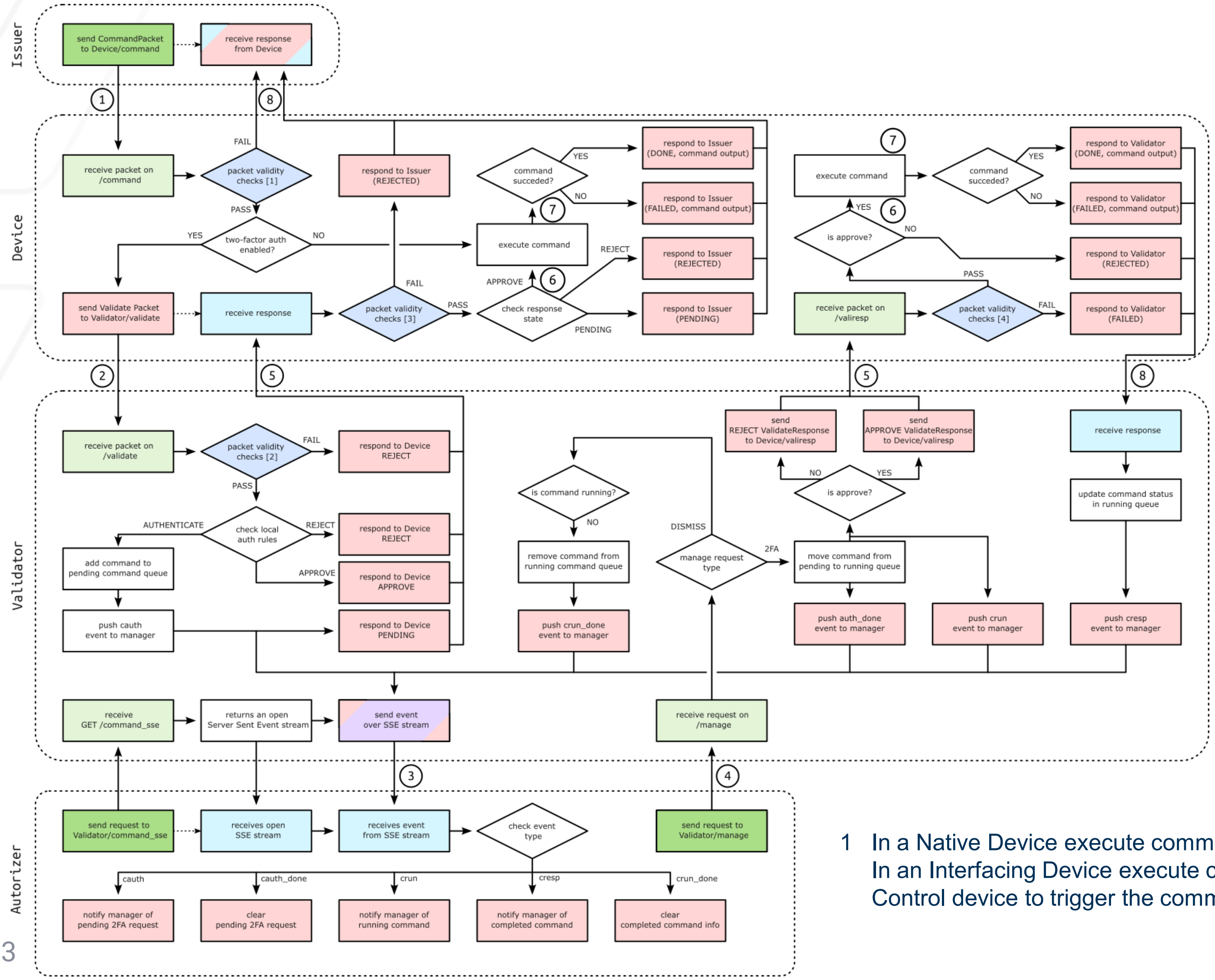
Security Details

- Password Hashing: scrypt key derivation function [12, 13]
- Packet Signing: RFC 2104 HMAC using SHA256 [7]
- SSL Encryption: 2048 bit RSA keys w/ self-signed certificates
- Server Stack:
  - Flask WSGI backend [9] -> uWSGI Server [10] -> NGINX Reverse Proxy [11]

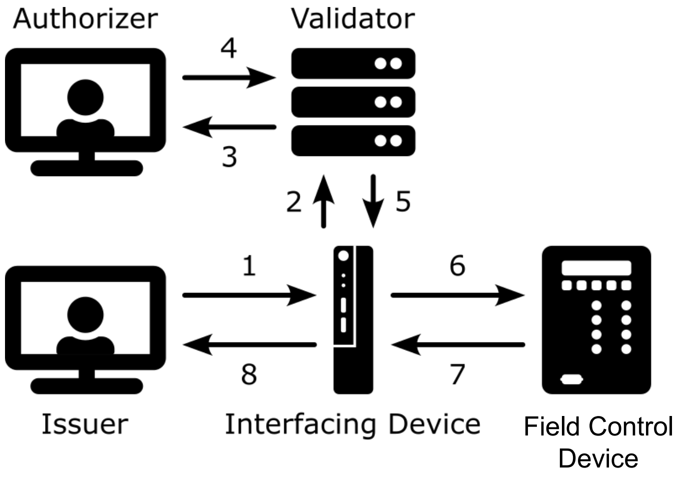
Codebase

python	6945
css	232
html	600
js	236
sql	46
total	8059

Edge Intercept Mode Control Flow



Issuer
send CommandPacket to Device/command
receive response from Device
Device
receive packet on /command
FAIL
packet validity checks [1]
PASS
respond to Issuer (REJECTED)
YES
two-factor auth enabled?
NO
execute command
command succeded?
YES
NO
respond to Issuer (DONE, command output)
respond to Issuer (FAILED, command output)
REJECT
respond to Issuer (REJECTED)
APPROVE
send Validate Packet to Validator/validate
receive response
packet validity checks [3]
FAIL
PASS
check response state
PENDING
respond to Issuer (PENDING)
execute command
command succeded?
YES
NO
respond to Validator (DONE, command output)
respond to Validator (FAILED, command output)
is approve?
YES
NO
respond to Validator (REJECTED)
PASS
receive packet on /valiresp
packet validity checks [4]
FAIL
respond to Validator (FAILED)
1
2
3
4
5
6
7
8
Validator
receive packet on /validate
packet validity checks [2]
FAIL
respond to Device REJECT
PASS
AUTHENTICATE
check local auth rules
REJECT
respond to Device REJECT
add command to pending command queue
APPROVE
respond to Device APPROVE
push cauth event to manager
respond to Device PENDING
is command running?
NO
remove command from running command queue
push crun_done event to manager
DISMISS
manage request type
2FA
send REJECT ValidateResponse to Device/valiresp
send APPROVE ValidateResponse to Device/valiresp
NO
YES
is approve?
move command from pending to running queue
push auth_done event to manager
push crun event to manager
receive response
update command status in running queue
push cresp event to manager
receive GET /command_sse
returns an open Server Sent Event stream
send event over SSE stream
receive request on /manage
Autorizer
send request to Validator/command_sse
receives open SSE stream
receives event from SSE stream
check event type
send request to Validator/manage
cauth
cauth_done
crun
cresp
crun_done
notify manager of pending 2FA request
clear pending 2FA request
notify manager of running command
notify manager of completed command
clear completed command info

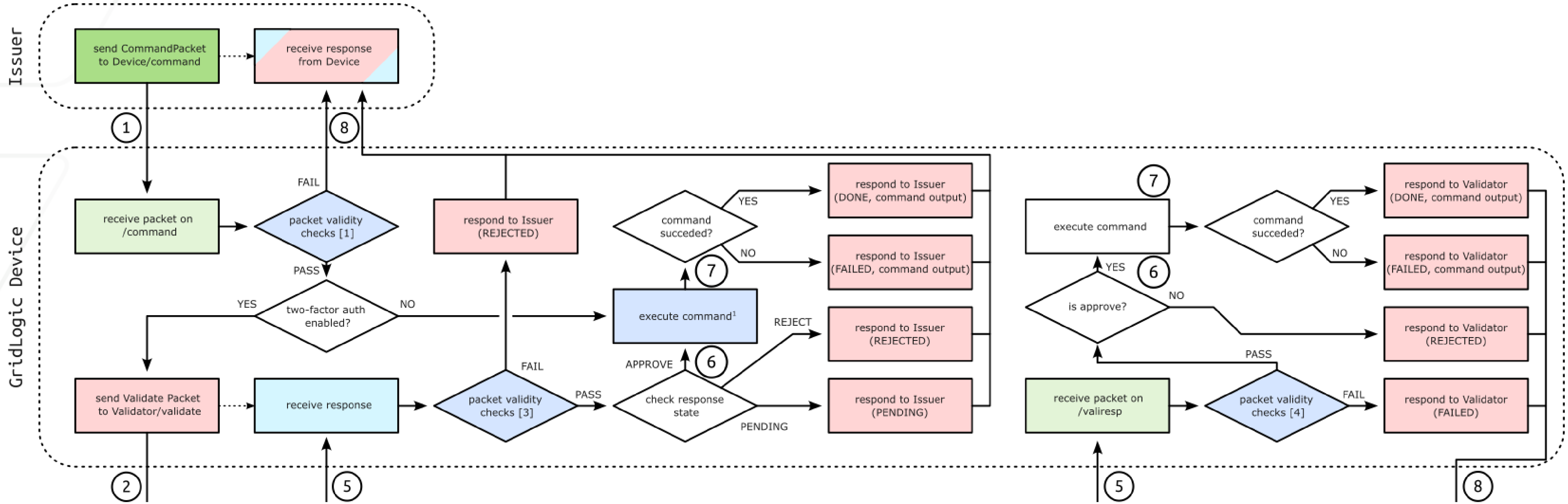


Legend

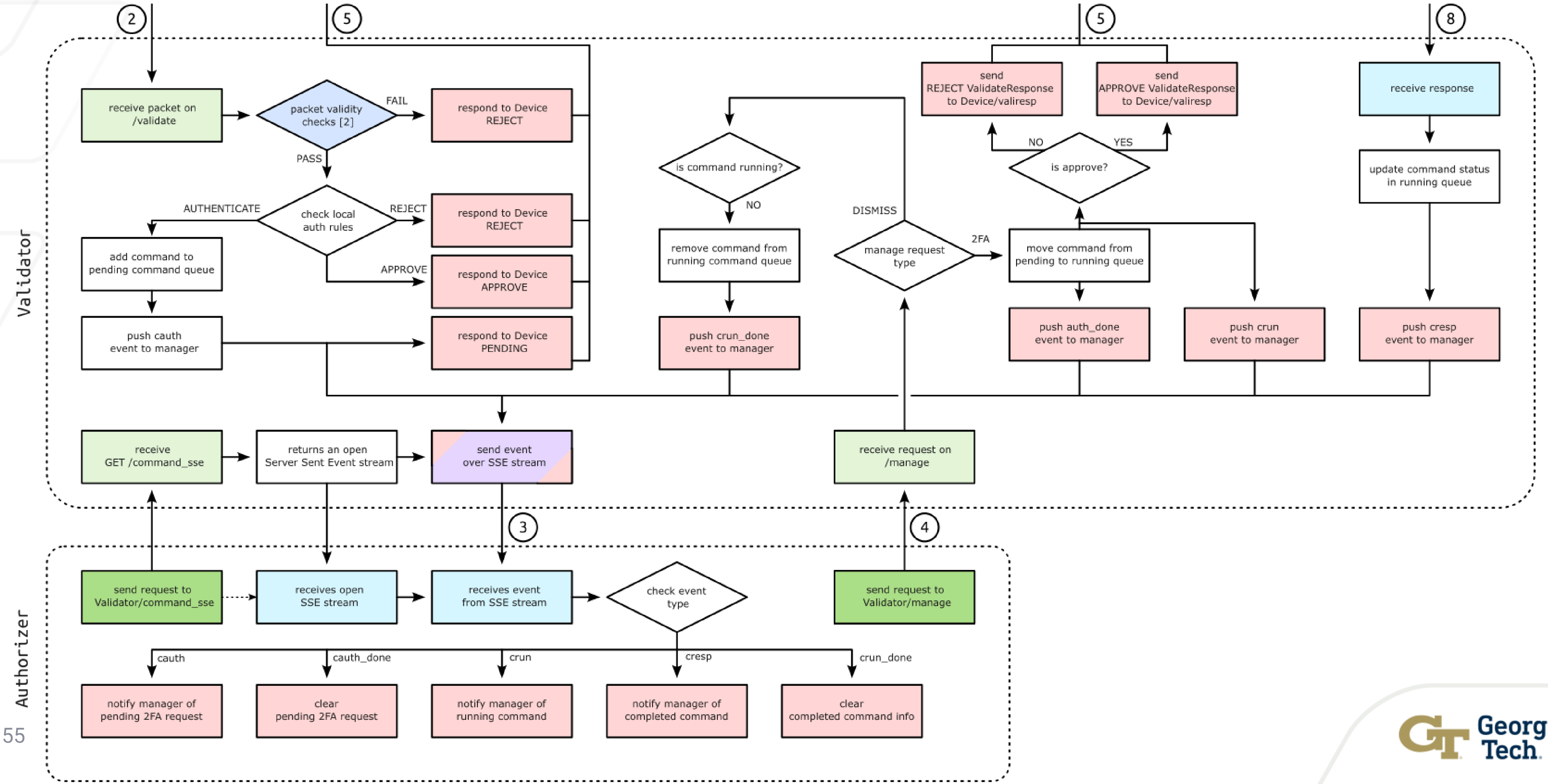
- 🟩 Entry Point (Human Trigger)
- 🟩 Control Flow Enters Block
- 🟦 Control Flow Returns to Block
- 🟥 Control Flow Exits Block
- 🟪 Control Flow Between Processes
- 🟦 Control Subgraph

[1] In a Native Device execute command directly executes the command (e.g., opens breaker). In an Interfacing Device execute command communicates over a secondary protocol to the Field Control device to trigger the command and waits for its completion.

Edge Intercept Mode Control Flow


Issuer
send CommandPacket to Device/command
receive response from Device
1
8
GridLogic Device
receive packet on /command
packet validity checks [1]
FAIL
PASS
respond to Issuer (REJECTED)
two-factor auth enabled?
YES
NO
command succeded?
YES
NO
respond to Issuer (DONE, command output)
respond to Issuer (FAILED, command output)
7
execute command¹
REJECT
respond to Issuer (REJECTED)
APPROVE
6
check response state
PENDING
respond to Issuer (PENDING)
send Validate Packet to Validator/validate
receive response
packet validity checks [3]
FAIL
PASS
2
5
7
execute command
command succeded?
YES
NO
respond to Validator (DONE, command output)
respond to Validator (FAILED, command output)
YES
6
is approve?
NO
respond to Validator (REJECTED)
PASS
receive packet on /valiresp
packet validity checks [4]
FAIL
respond to Validator (FAILED)
5
8

Edge Intercept Mode Control Flow



Validator

2

receive packet on /validate

packet validity checks [2]

FAIL

PASS

respond to Device REJECT

5

AUTHENTICATE

check local auth rules

REJECT

respond to Device REJECT

add command to pending command queue

APPROVE

respond to Device APPROVE

push cauth event to manager

respond to Device PENDING

is command running?

NO

remove command from running command queue

push crun_done event to manager

DISMISS

manage request type

2FA

5

send REJECT ValidateResponse to Device/valiresp

send APPROVE ValidateResponse to Device/valiresp

NO

YES

is approve?

move command from pending to running queue

push auth_done event to manager

push crun event to manager

8

receive response

update command status in running queue

push cresp event to manager

receive GET /command_sse

returns an open Server Sent Event stream

send event over SSE stream

receive request on /manage

3

4

Authorizer

send request to Validator/command_sse

receives open SSE stream

receives event from SSE stream

check event type

send request to Validator/manage

cauth

cauth_done

crun

cresp

crun_done

notify manager of pending 2FA request

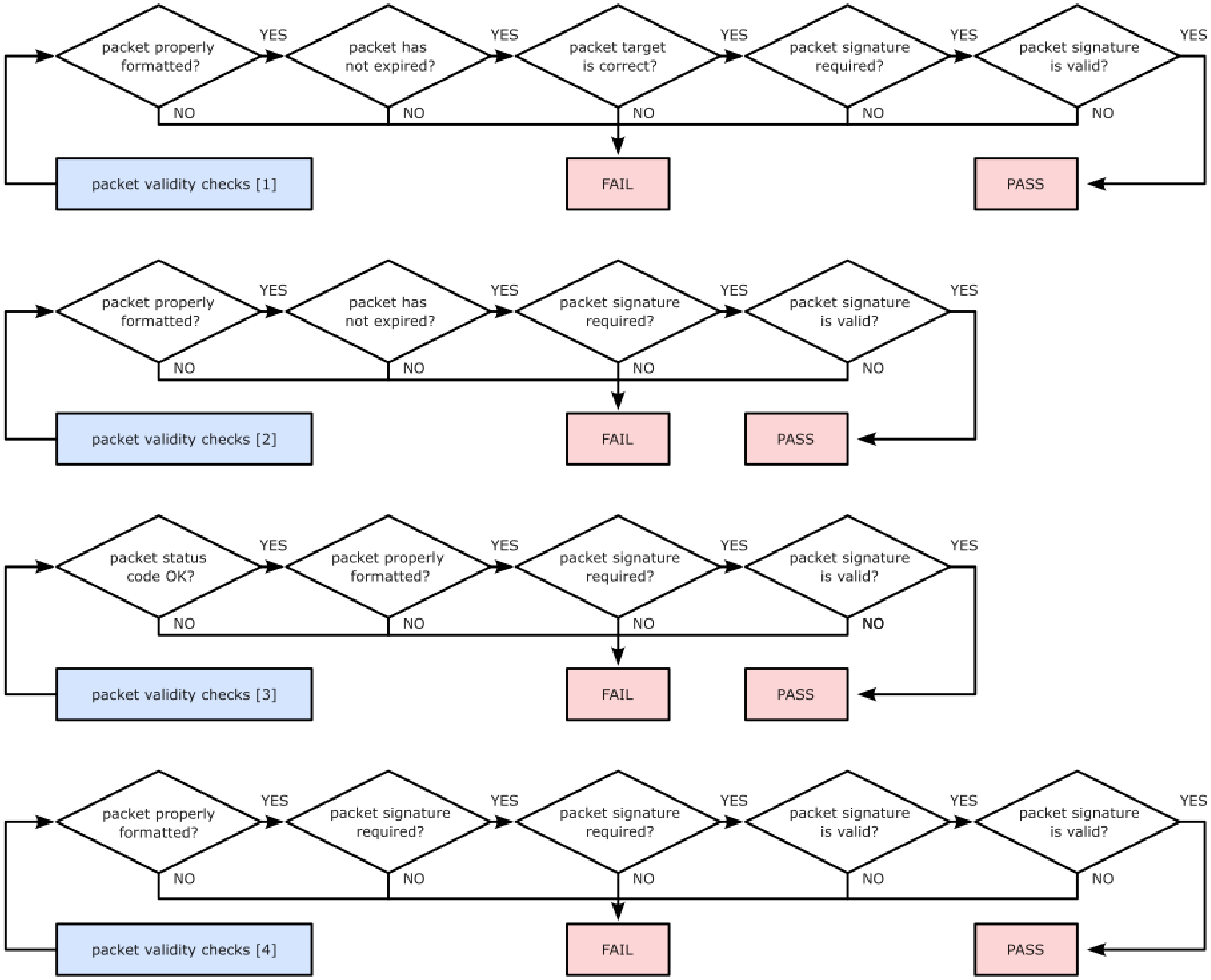
clear pending 2FA request

notify manager of running command

notify manager of completed command

clear completed command info

Edge Intercept Mode Control Flow



packet validity checks [1]

packet properly formatted? YES → packet has not expired? YES → packet target is correct? YES → packet signature required? YES → packet signature is valid? YES → PASS

NO → FAIL

packet validity checks [2]

packet properly formatted? YES → packet has not expired? YES → packet signature required? YES → packet signature is valid? YES → PASS

NO → FAIL

packet validity checks [3]

packet status code OK? YES → packet properly formatted? YES → packet signature required? YES → packet signature is valid? YES → PASS

NO → FAIL

packet validity checks [4]

packet properly formatted? YES → packet signature required? YES → packet signature required? YES → packet signature is valid? YES → packet signature is valid? YES → PASS

NO → FAIL