U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT COMMAND

Collaboration with Academia to Support Future Force Capabilities
Explore Computer Science Research

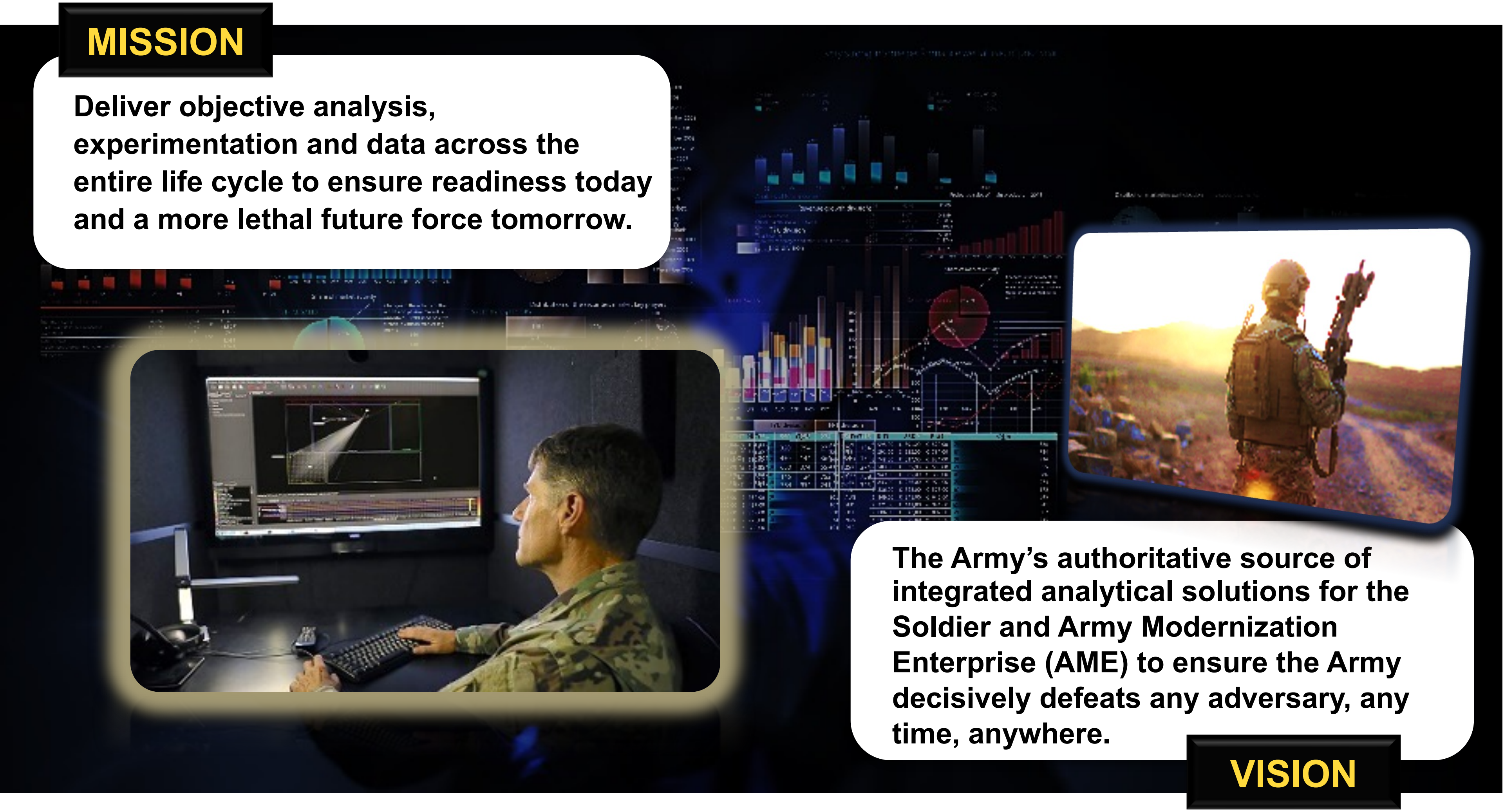
Cyber Experimentation & Analysis Division (CEAD)

Dr. Oscar Perez and Dr. Jayashree Harikumar

DEVCOM Analysis Center

Nov 4, 2022

DAC MISSION AND VISION



MISSION

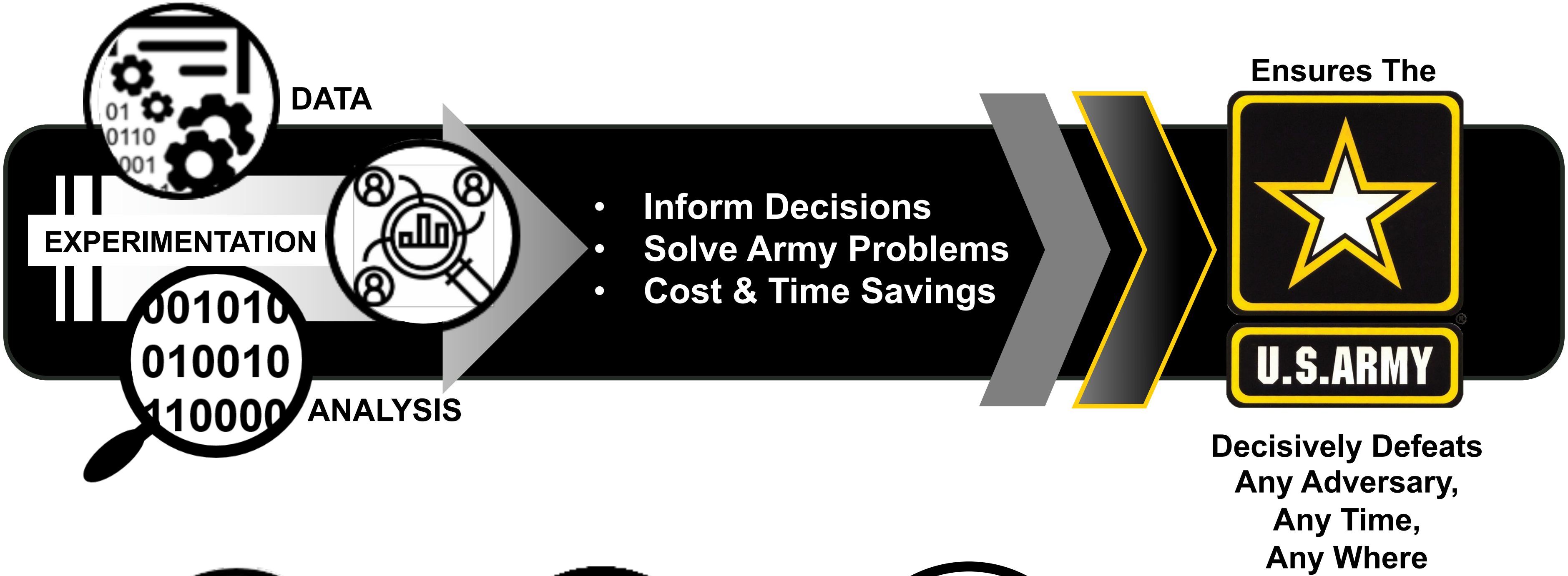
Deliver objective analysis, experimentation and data across the entire life cycle to ensure readiness today and a more lethal future force tomorrow.

The Army's authoritative source of integrated analytical solutions for the Soldier and Army Modernization Enterprise (AME) to ensure the Army decisively defeats any adversary, any time, anywhere.

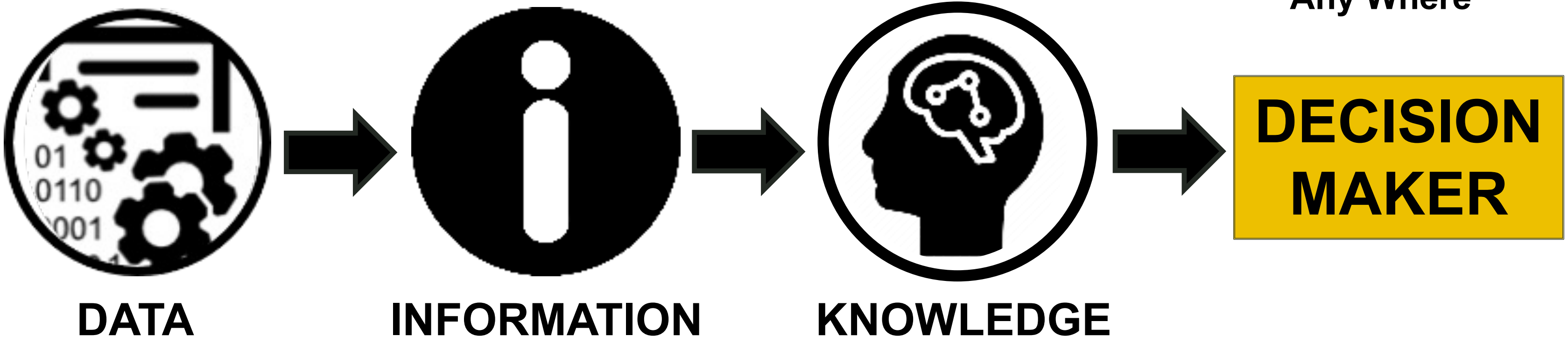
VISION

TODAY'S ANALYTICS FOR TOMORROW'S SOLDIER - FORGE THE FUTURE

HOW WE MEET OUR MISSION



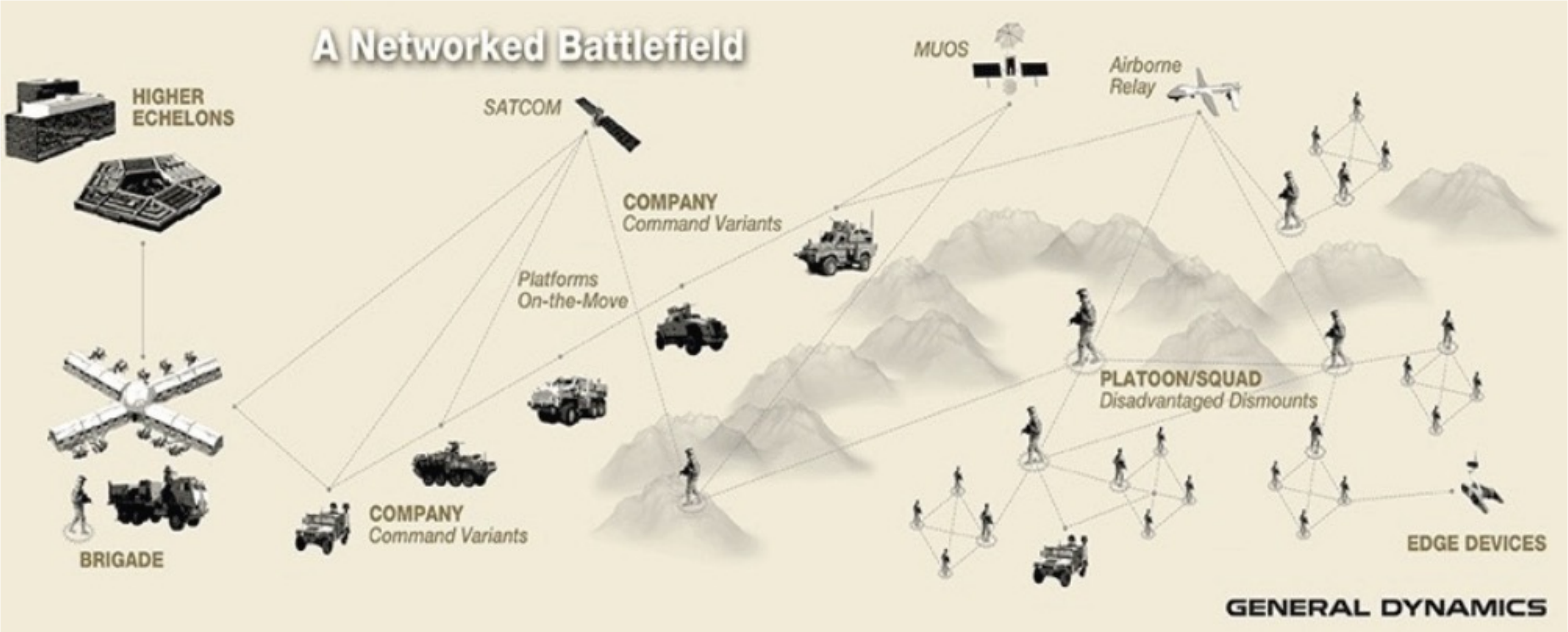
- Inform Decisions
- Solve Army Problems
- Cost & Time Savings



People - Readiness - Modernization

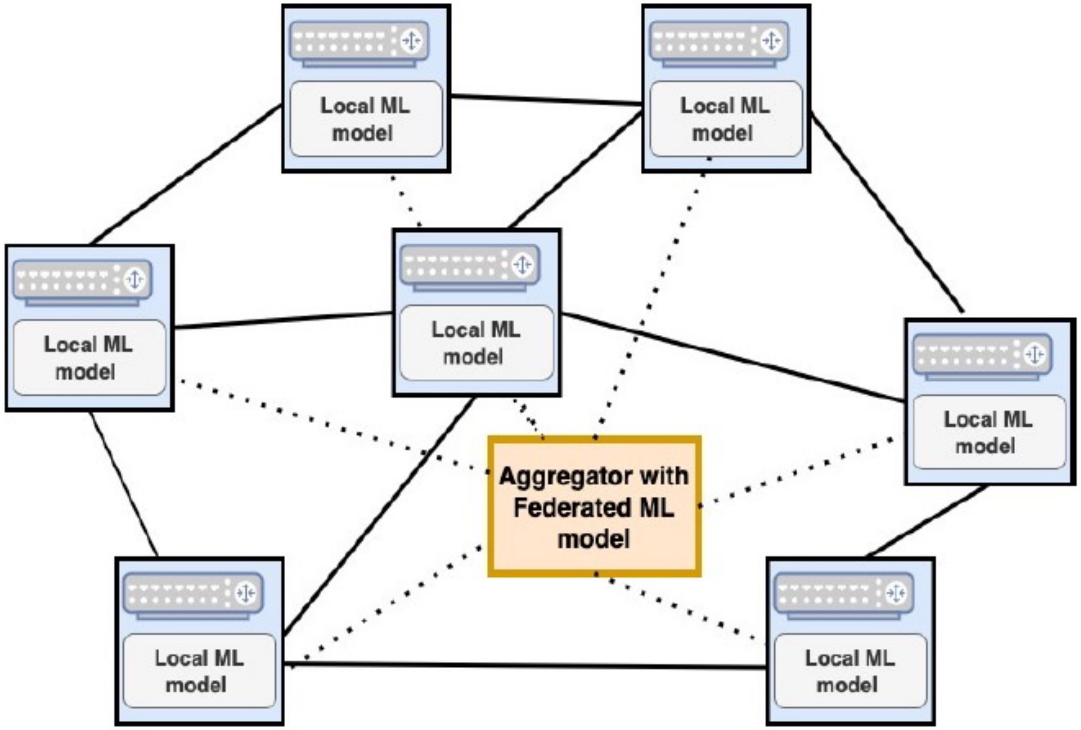
USING MACHINE LEARNING TO PROTECT NETWORKS

- Collaborative Efforts with NMSU
  - Federated ML for network management and vulnerability assessment
    - Using averaging at the aggregator to discern potential attack scenarios at the individual routers via assessing divergence in model parameters sent by the individual routers
  - Probing attack on Networks and mitigation using ML
    - machine-learning empowered automation tool to identify and disable the suspicious probing packet forwarding



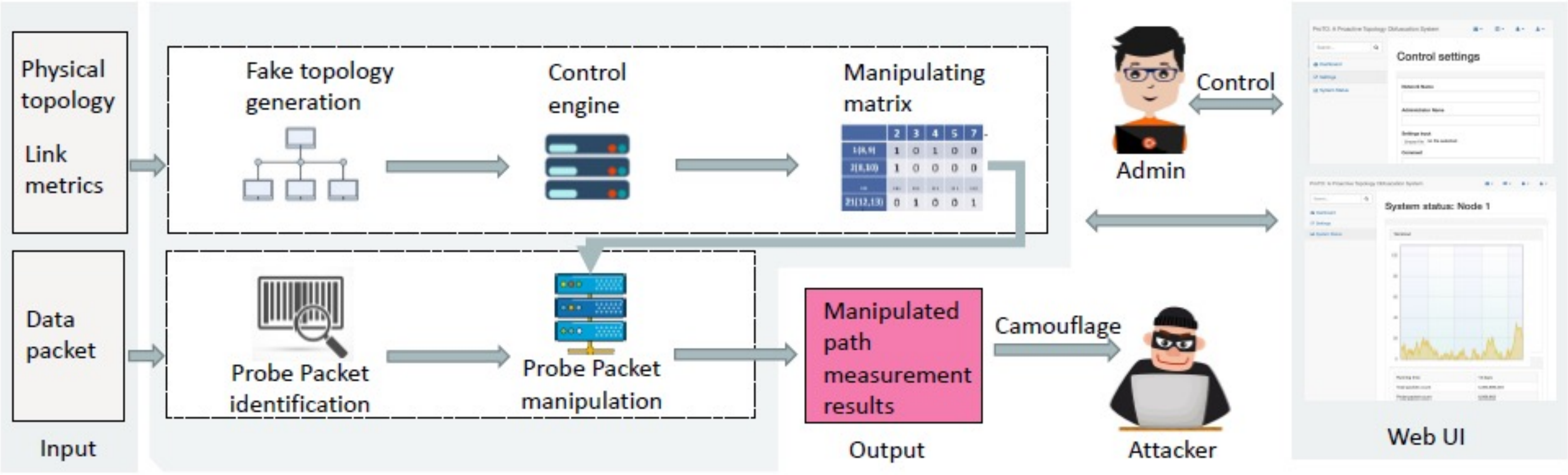
USING MACHINE LEARNING TO PROTECT NETWORKS

- Collaborative Efforts with NMSU: **Federated ML for network management and vulnerability assessment**
  - Local ML models at the routers learn from their local observations and share their model abstracts/parameters with an aggregator.
  - The aggregator performs federated averaging to create a global model that encompasses the global state of the network.



USING MACHINE LEARNING TO PROTECT NETWORKS

- Collaborative Efforts with NMSU: **Probing attack on Networks and mitigation using ML**
  - Understanding and evaluating existing end-to-end external topology inferring strategies
  - Designing a reliable and secure machine-learning based probing behavior identifier and testing its performance
  - Developing and testing a topology obfuscation mechanism to mitigate probing attacks
  - Comprehensively evaluating and testing the efficiency and effectiveness of the proposed automation tool



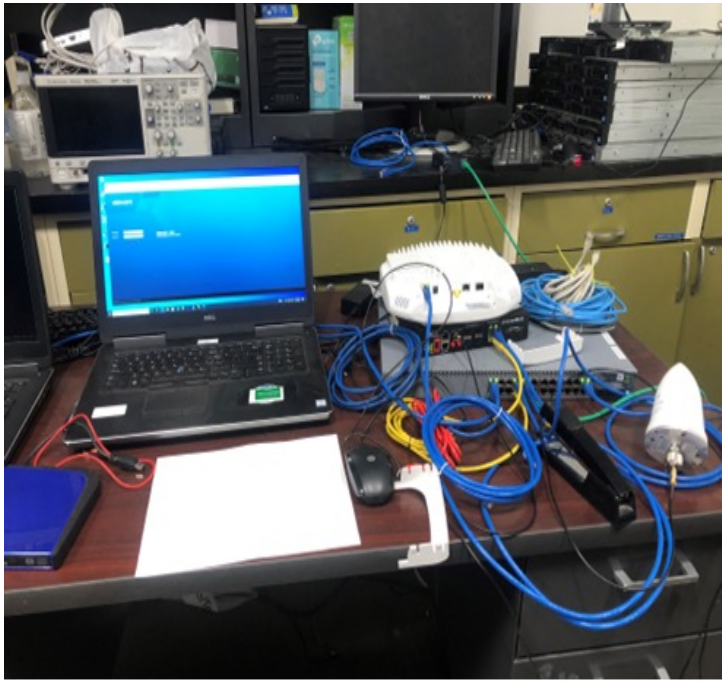
	2	3	4	5	7
1(8,9)	1	0	1	0	0
2(8,10)	1	0	0	0	0
...	...	...	...	...	...
21(12,13)	0	1	0	0	1

CEMA AND CELLULAR COMMUNICATIONS

- What is CEMA?
- What is cellular communications?
- Why will traditional cyber analysis methods not work?
- Testbeds methodology and development to test future-G technologies
  - Experiment setup
  - Equipment specification and cross pollination
  - Troubleshooting

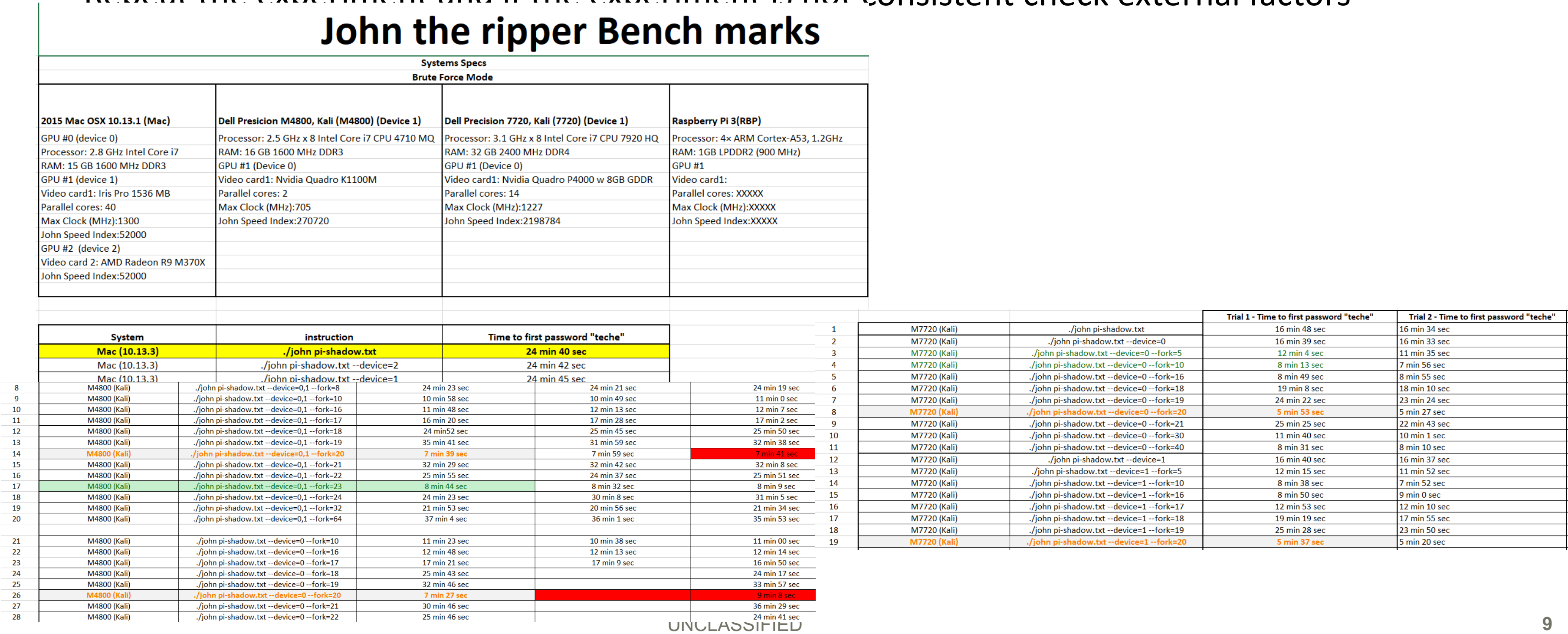
CELLULAR COMMUNICATIONS COLLABORATION

- To date we have built a 4G and LTE testbed to study the impact of
  - Cyber activities on EW affected systems and vice versa
  - Combined effect of EW and Cyber activities on cellular systems
- Analysis of the cellular network uses an innovative CEMA approach
- Possible DoD support for future Army 5G Cyber research
- Academia:
  - Advantages (faculty and student participation; student development)
  - Disadvantages (Information classification)
- Industry:
  - Advantages (availability of ready or near-ready tools)
  - Disadvantages (license, cost,..)



PERFORMANCE OPTIMIZATION RESEARCH FOR PARALLEL PROCESSING USING GPUS

- Optimization on password cracking based on hardware and GPU availability
  - Create a base line and get metric on your controlled environment
    - Make sure your base line is constant, do not take only one measurement
  - Introduce one variable and analyze the effect
    - After one variable has been analyzed introduce a second variable
  - Repeat the experiment and if the experiment is not consistent check external factors



John the ripper Bench marks

Systems Specs Brute Force Mode			
2015 Mac OSX 10.13.1 (Mac)	**Dell Presicion M4800, Kali (M4800) (Device 1)**	**Dell Precision 7720, Kali (7720) (Device 1)**	**Raspberry Pi 3(RBP)**
GPU #0 (device 0)	Processor: 2.5 GHz x 8 Intel Core i7 CPU 4710 MQ	Processor: 3.1 GHz x 8 Intel Core i7 CPU 7920 HQ	Processor: 4× ARM Cortex-A53, 1.2GHz
Processor: 2.8 GHz Intel Core i7	RAM: 16 GB 1600 MHz DDR3	RAM: 32 GB 2400 MHz DDR4	RAM: 1GB LPDDR2 (900 MHz)
RAM: 15 GB 1600 MHz DDR3	GPU #1 (Device 0)	GPU #1 (Device 0)	GPU #1
GPU #1 (device 1)	Video card1: Nvidia Quadro K1100M	Video card1: Nvidia Quadro P4000 w 8GB GDDR	Video card1:
Video card1: Iris Pro 1536 MB	Parallel cores: 2	Parallel cores: 14	Parallel cores: XXXXX
Parallel cores: 40	Max Clock (MHz):705	Max Clock (MHz):1227	Max Clock (MHz):XXXXX
Max Clock (MHz):1300	John Speed Index:270720	John Speed Index:2198784	John Speed Index:XXXXX
John Speed Index:52000			
GPU #2 (device 2)			
Video card 2: AMD Radeon R9 M370X			
John Speed Index:52000			

System	instruction	Time to first password "teche"
Mac (10.13.3)	**./john pi-shadow.txt**	**24 min 40 sec**
Mac (10.13.3)	./john pi-shadow.txt --device=2	24 min 42 sec
Mac (10.13.3)	./john pi-shadow.txt --device=1	24 min 45 sec

8	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=8	24 min 23 sec	24 min 21 sec	24 min 19 sec
9	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=10	10 min 58 sec	10 min 49 sec	11 min 0 sec
10	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=16	11 min 48 sec	12 min 13 sec	12 min 7 sec
11	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=17	16 min 20 sec	17 min 28 sec	17 min 2 sec
12	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=18	24 min52 sec	25 min 45 sec	25 min 50 sec
13	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=19	35 min 41 sec	31 min 59 sec	32 min 38 sec
14	**M4800 (Kali)**	**./john pi-shadow.txt --device=0,1 --fork=20**	**7 min 39 sec**	7 min 59 sec	7 min 41 sec
15	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=21	32 min 29 sec	32 min 42 sec	32 min 8 sec
16	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=22	25 min 55 sec	24 min 37 sec	25 min 51 sec
17	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=23	8 min 44 sec	8 min 32 sec	8 min 9 sec
18	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=24	24 min 23 sec	30 min 8 sec	31 min 5 sec
19	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=32	21 min 53 sec	20 min 56 sec	21 min 34 sec
20	M4800 (Kali)	./john pi-shadow.txt --device=0,1 --fork=64	37 min 4 sec	36 min 1 sec	35 min 53 sec
21	M4800 (Kali)	./john pi-shadow.txt --device=0 --fork=10	11 min 23 sec	10 min 38 sec	11 min 00 sec
22	M4800 (Kali)	./john pi-shadow.txt --device=0 --fork=16	12 min 48 sec	12 min 13 sec	12 min 14 sec
23	M4800 (Kali)	./john pi-shadow.txt --device=0 --fork=17	17 min 21 sec	17 min 9 sec	16 min 50 sec
24	M4800 (Kali)	./john pi-shadow.txt --device=0 --fork=18	25 min 43 sec		24 min 17 sec
25	M4800 (Kali)	./john pi-shadow.txt --device=0 --fork=19	32 min 46 sec		33 min 57 sec
26	**M4800 (Kali)**	**./john pi-shadow.txt --device=0 --fork=20**	**7 min 27 sec**		9 min 8 sec
27	M4800 (Kali)	./john pi-shadow.txt --device=0 --fork=21	30 min 46 sec		36 min 29 sec
28	M4800 (Kali)	./john pi-shadow.txt --device=0 --fork=22	25 min 46 sec		24 min 41 sec

			Trial 1 - Time to first password "teche"	Trial 2 - Time to first password "teche"
1	M7720 (Kali)	./john pi-shadow.txt	16 min 48 sec	16 min 34 sec
2	M7720 (Kali)	./john pi-shadow.txt --device=0	16 min 39 sec	16 min 33 sec
3	M7720 (Kali)	./john pi-shadow.txt --device=0 --fork=5	12 min 4 sec	11 min 35 sec
4	M7720 (Kali)	./john pi-shadow.txt --device=0 --fork=10	8 min 13 sec	7 min 56 sec
5	M7720 (Kali)	./john pi-shadow.txt --device=0 --fork=16	8 min 49 sec	8 min 55 sec
6	M7720 (Kali)	./john pi-shadow.txt --device=0 --fork=18	19 min 8 sec	18 min 10 sec
7	M7720 (Kali)	./john pi-shadow.txt --device=0 --fork=19	24 min 22 sec	23 min 24 sec
8	**M7720 (Kali)**	**./john pi-shadow.txt --device=0 --fork=20**	**5 min 53 sec**	5 min 27 sec
9	M7720 (Kali)	./john pi-shadow.txt --device=0 --fork=21	25 min 25 sec	22 min 43 sec
10	M7720 (Kali)	./john pi-shadow.txt --device=0 --fork=30	11 min 40 sec	10 min 1 sec
11	M7720 (Kali)	./john pi-shadow.txt --device=0 --fork=40	8 min 31 sec	8 min 10 sec
12	M7720 (Kali)	./john pi-shadow.txt --device=1	16 min 40 sec	16 min 37 sec
13	M7720 (Kali)	./john pi-shadow.txt --device=1 --fork=5	12 min 15 sec	11 min 52 sec
14	M7720 (Kali)	./john pi-shadow.txt --device=1 --fork=10	8 min 38 sec	7 min 52 sec
15	M7720 (Kali)	./john pi-shadow.txt --device=1 --fork=16	8 min 50 sec	9 min 0 sec
16	M7720 (Kali)	./john pi-shadow.txt --device=1 --fork=17	12 min 53 sec	12 min 10 sec
17	M7720 (Kali)	./john pi-shadow.txt --device=1 --fork=18	19 min 19 sec	17 min 55 sec
18	M7720 (Kali)	./john pi-shadow.txt --device=1 --fork=19	25 min 28 sec	23 min 50 sec
19	**M7720 (Kali)**	**./john pi-shadow.txt --device=1 --fork=20**	**5 min 37 sec**	5 min 20 sec

QUESTIONS