Reconstructing Alert Trees for Cyber Triage

Eric Ficke

PhD Dissertation Defense

The University of Texas at San Antonio

April 13, 2022

Committee Co-Chairs

Prof. Ravi Sandhu

Prof. Shouhuai Xu

Committee Members

Prof. Xiaoyin Wang

Prof. Greg White

Prof. Mimi Xie

UTSA®

Publications

- Published
  1. S. He, **E. Ficke**, M. Pritom, H. Chen, Q. Tang, Q. Chen, M. Pendleton, L. Njilla, and S. Xu. *Blockchain-based Automated and Robust Cyber Security Management*, Journal of Parallel Distributed Computing, 163: 62-82 (2022)
  2. **E. Ficke** and S. Xu. APIN: *Automatic Attack Path Identification in Computer Networks*, IEEE International Conference on Intelligence and Security Informatics (ISI), 2020. **[Dissertation Chapter 2]**
  3. **E. Ficke**, K. Schweitzer, R. Bateman, and S. Xu. *Analyzing Root Causes of Intrusion Detection False-Negatives: Methodology and Case Study*. IEEE Military Communications Conference (MILCOM), 2019.
  4. J. Mireles, **E. Ficke**, J. Cho, P. Hurley, and S. Xu. *Metrics Towards Measuring Cyber Agility*. IEEE Transactions on Information Forensics and Security (IEEE T-IFS), 14(12): 3217-3232 (2019).
  5. **E. Ficke**, K. Schweitzer, R. Bateman, and S. Xu. *Characterizing the Effectiveness of Network-Based Intrusion Detection Systems*. IEEE Military Communications Conference (MILCOM), 2018.
- Manuscripts to be submitted for review
  1. **E. Ficke**, R. Bateman, and S. Xu. AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees. **[Dissertation Chapter 3]**
  2. R. Garcia-LeBron, **E. Ficke**, W. Wu, S. Xu. Characterizing Cyber Attack Reconnaissance Trajectories.
  3. **E. Ficke**, R. Bateman, and S. Xu. Alert Tree Reduction and Visualization. **[Dissertation Chapter 4]**

UTSA®

Dissertation Outline

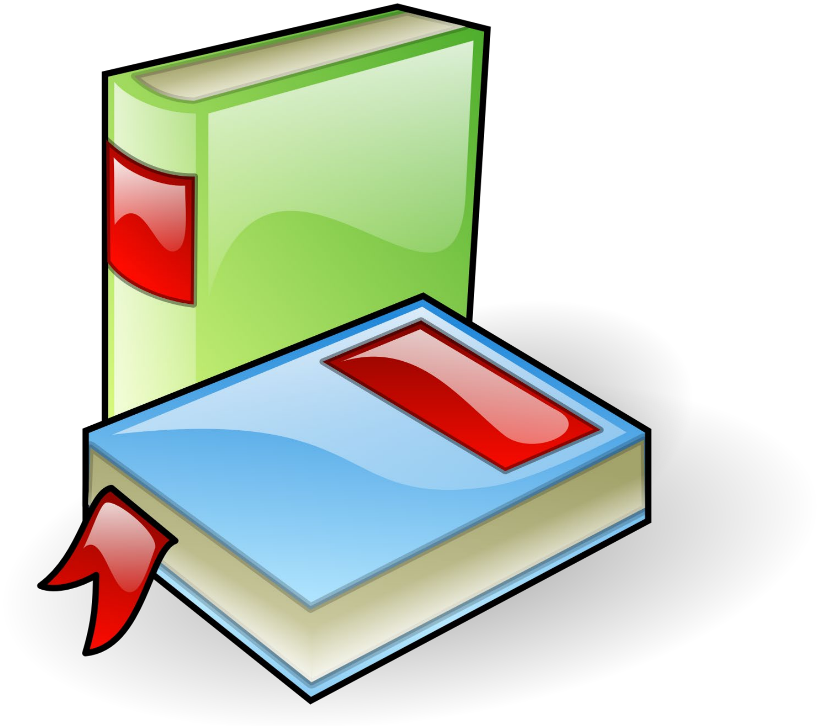
- Introduction
- APIN: Alert Path Identification in Computer Networks
- AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees
- Alert Tree Reduction and Visualization
- Conclusion

Introduction

Chapter 1

Background – Alert Trees

- Cyber Triage (Network-level)
  - Alert prioritization
  - Alert correlation
  - Attack lifecycle
- Attack Prediction
  - Attack graphs / trees / paths
  - Vulnerability graphs



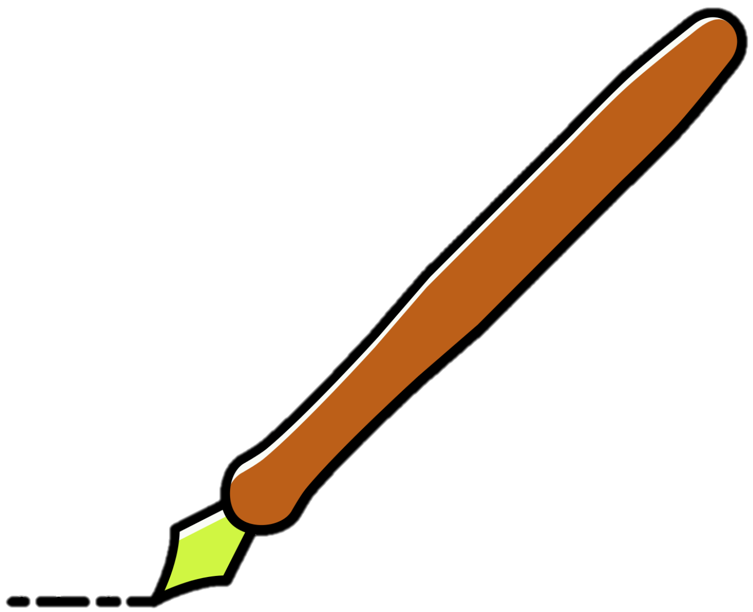
Motivation

- Alert volume
  - Unrealistically low in ad hoc datasets
  - **Overwhelms human analysis** in real data
- Alert graph / tree / path formalization
  - Varies by usage
  - Depends on **spatial and temporal dependencies**

UTSA

Chapter Themes

2. Alert Path Identification (**APIN**)
    - Alert path reconstruction
    - Threat score (TS) ranking
3. Cumulative Reconstruction (**AutoCRAT**)
    - Alert tree reconstruction
    - Alternative path reconstruction method
    - Asymptotic and real analysis
4. Reduction and Visualization
    - Mitigates emergent problem of tree size



UTSA®

Dissertation Outline

- Introduction
- APIN: Alert Path Identification in Computer Networks
- AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees
- Alert Tree Reduction and Visualization
- Conclusion

APIN: Alert Path Identification in Computer Networks

Chapter 2

Motivation: Cyber Triage

- Time sensitive
- Resource intensive
- Error prone
- Large search space

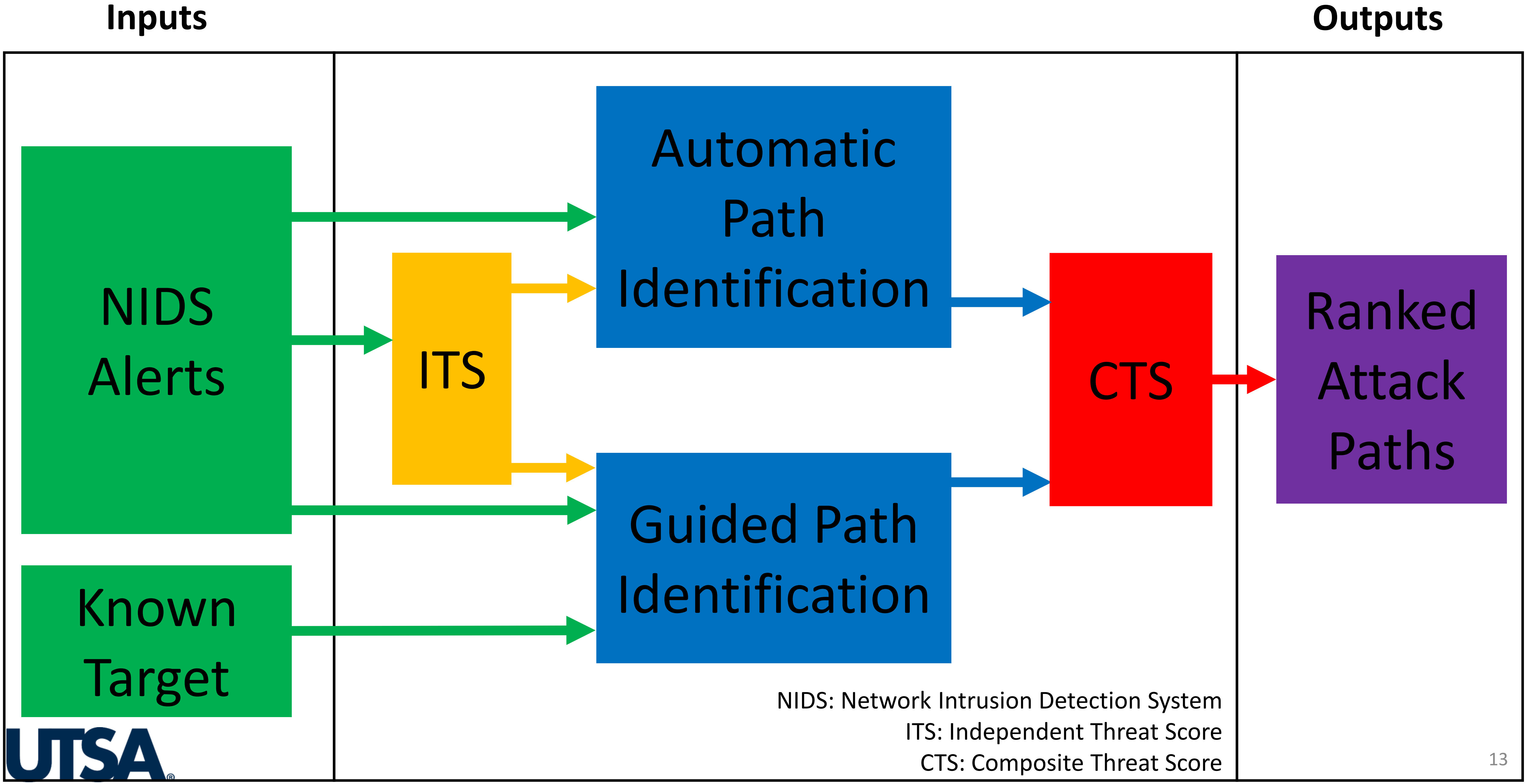
UTSA

Contributions

- Attack Tracking
  - Alert paths show footprints between victim computers
  - Spatiotemporal path reconstruction method
- Heuristics
  - Threat score shows attacker effort
  - (Actual compromise may vary)

UTSA

APIN Framework



NIDS: Network Intrusion Detection System
ITS: Independent Threat Score
CTS: Composite Threat Score

Metric: Independent Threat Score

- Input:
  - A_{in} alert types (inbound)
  - A_{out} alert types (outbound)
- Terms:
  - $D_{in} = |A_{in}|$
  - $D_{out} = |A_{out}|$
  - $S_{in} = \sqrt[|A_{in}|]{\prod_{a \in A_{in}} |a|}$

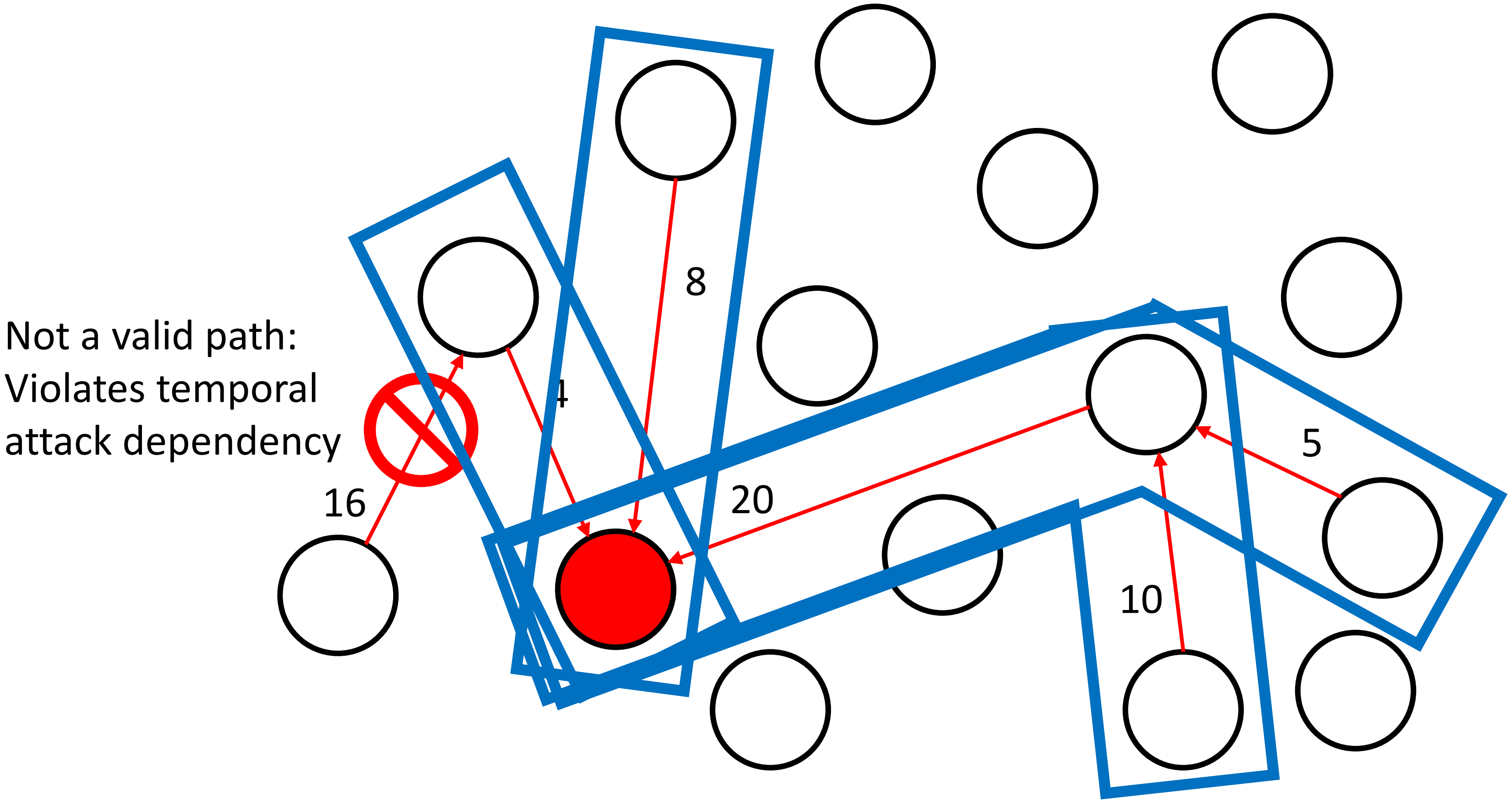
$$\textbf{ITS} = \sqrt[3]{D_{in} \cdot D_{out} \cdot S_{in}}$$

"D" represents alert diversity
"S" represents alert scale (by type)

UTSA

Methods (Alert Path Identification)

Approach: breadth-first search in reverse-chronological order



Edges indexed by timestamp

Preliminary Analysis

- Scans (high volume, low threat)
  - …ritize inbound alerts
- …r threat)
  - Cau…
  - Blacklist nodes
    - Restricts path identification
    - Leaves nodes unmonitored

High-granularity network segmentation improves performance significantly

Metric: Weighted Independent Threat Score

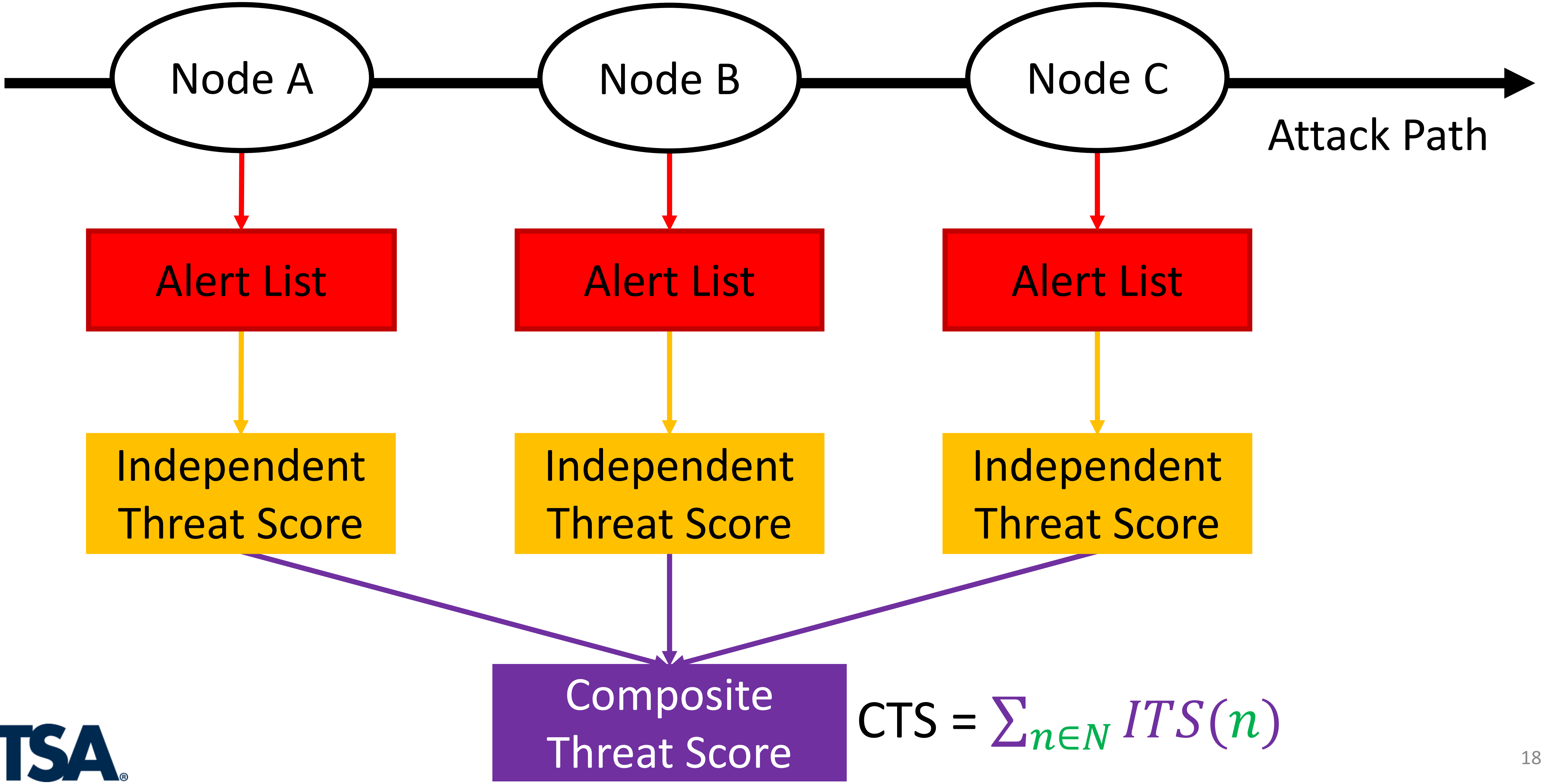
- Input:
  - A_{in} alert types (inbound)
  - A_{out} alert types (outbound)
- Terms:
  - $D_{in} = |A_{in}|$
  - $D_{out} = |A_{out}|$
  - $S_{in} = \sqrt[|A_{in}|]{\prod_{a \in A_{in}} |a|}$
  - $W = w_1 + w_2 + w_3$

$$ITS = \sqrt[W]{D_{in}^{w_1} \cdot D_{out}^{w_2} \cdot S_{in}^{w_3}}$$

"D" represents alert diversity
"S" represents alert scale (by type)

UTSA®

Metric: Composite Threat Score



$$CTS = \sum_{n \in N} ITS(n)$$

Preliminary Results: DARPA ’99

- Notable paths, using queries from top 5 nodes

Artificial datasets need multi-step attacks

[illegible] Origin	Composite Threat Score ∇	Length (#edges)	Notable Alerts
[illegible] 209.67.29.11	[illegible]	1	Windows 95 Malware
172.16.116.194	3.43	[illegible]	[illegible]
 207.25.71.141	3.41	1	Windows 95 [illegible]
192.168.1.30 172.16.112.5	3.41	1	Public SNMP Access
 206.132.25.51	3.36	1	Windows 95 Malware

Hidden IPs are repeated from higher-ranked paths

UTSA

Results: CSE-CIC-IDS2018

[Path Origin, Path Target]	Composite Threat Score ∇	Length (#Edges)	Notable Alerts
103.47.124.154 54.172.47.69	34.31	4	EternalBlue (WannaCry) NAT Traversal
172.31.67.54 52.87.201.4	33.60	3	EternalBlue (WannaCry) NAT Traversal
71.6.165.200 172.31.64.78	26.42	3	Blacklisted IP group SQL Scan
77.222.106.20 172.31.66.112	21.30	3	EternalBlue (WannaCry) SMB Share Access
172.31.64.78 172.31.0.2	21.11	1	Suspicious DNS Query

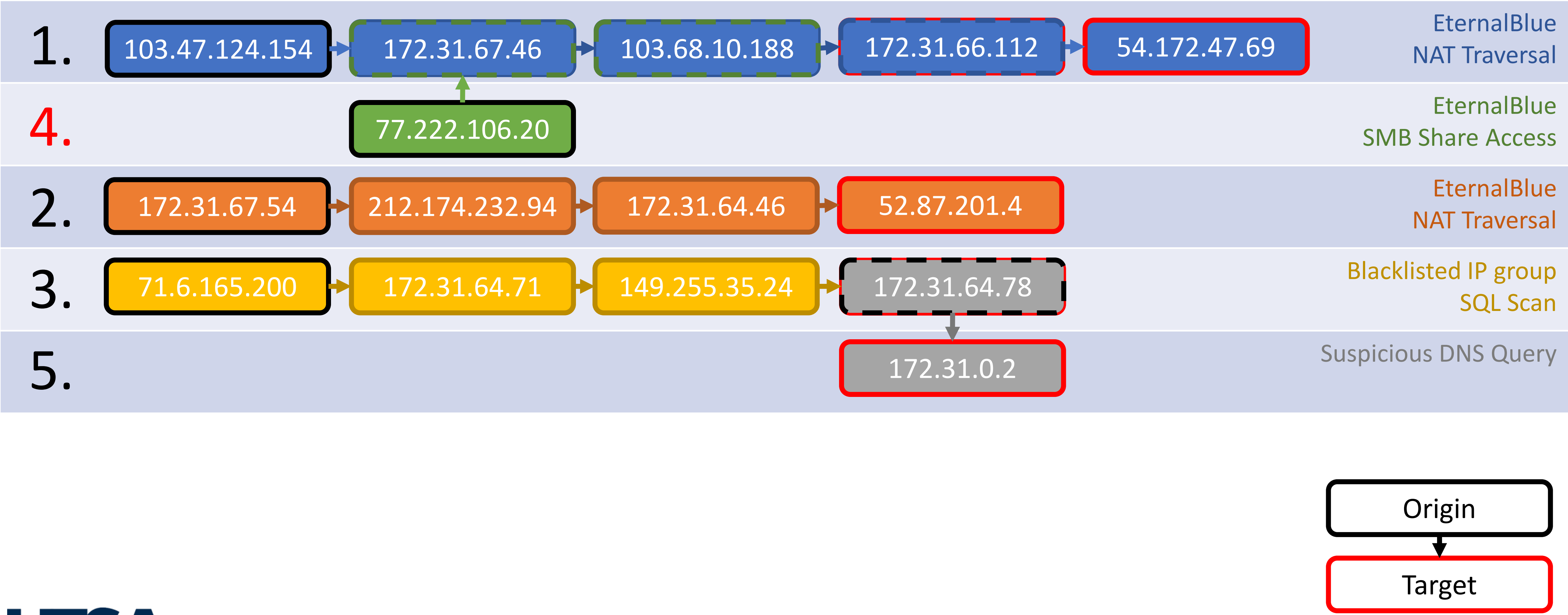
UTSA®

Results: CSE-CIC-IDS2018

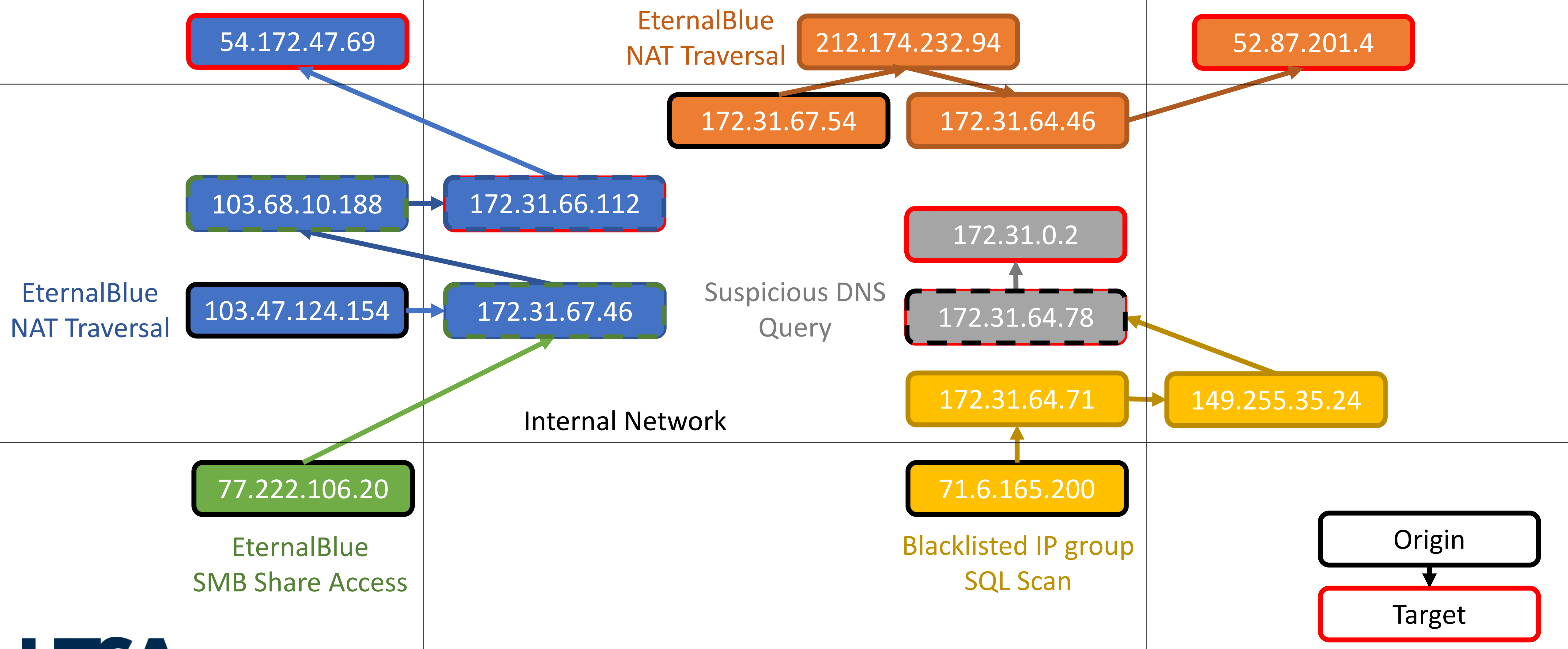
1.	103.47.124.154 → 172.31.67.46 → 103.68.10.188 → 172.31.66.112 → 54.172.47.69	EternalBlue NAT Traversal
2.	172.31.67.54 → 212.174.232.94 → 172.31.64.46 → 52.87.201.4	EternalBlue NAT Traversal
3.	71.6.165.200 → 172.31.64.71 → 149.255.35.24 → 172.31.64.78	Blacklisted IP group SQL Scan
4.	77.222.106.20 → 172.31.67.46 → 103.68.10.188 → 172.31.66.112	EternalBlue SMB Share Access
5.	172.31.64.78 → 172.31.0.2	Suspicious DNS Query

UTSA

Results: CSE-CIC-IDS2018



Results: CSE-CIC-IDS2018



Dissertation Outline

- Introduction
- APIN: Alert Path Identification in Computer Networks
- AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees
- Alert Tree Reduction and Visualization
- Conclusion

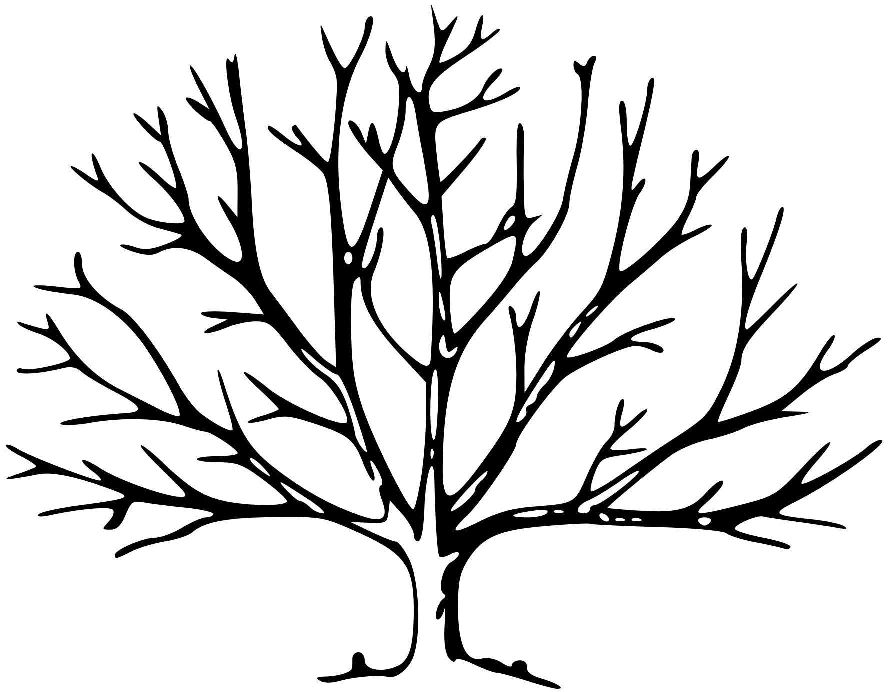
AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees

Chapter 3

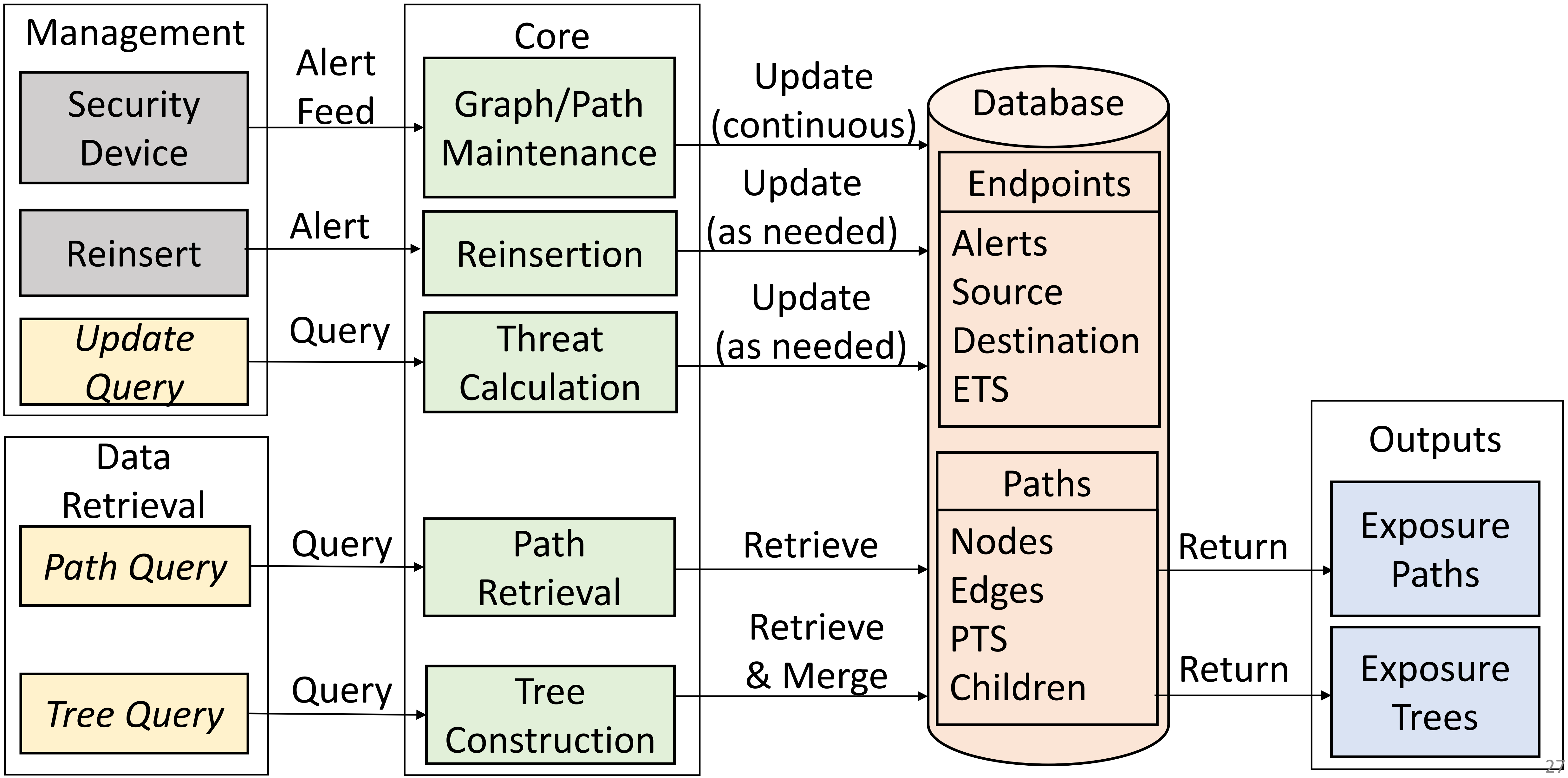
UTSA®

Motivation: Alert Tree Optimization

- Improve reconstruction
- Identify optimization tradeoffs
- Formalize alert trees

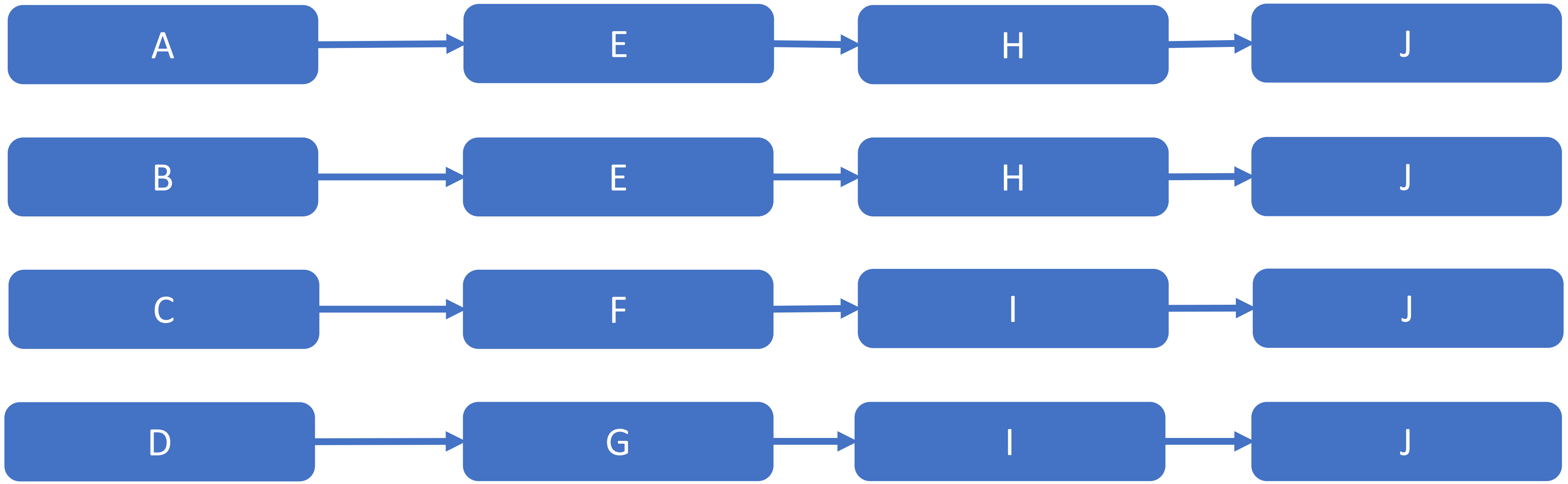


AutoCRAT Architecture



Methods (Path Maintenance)

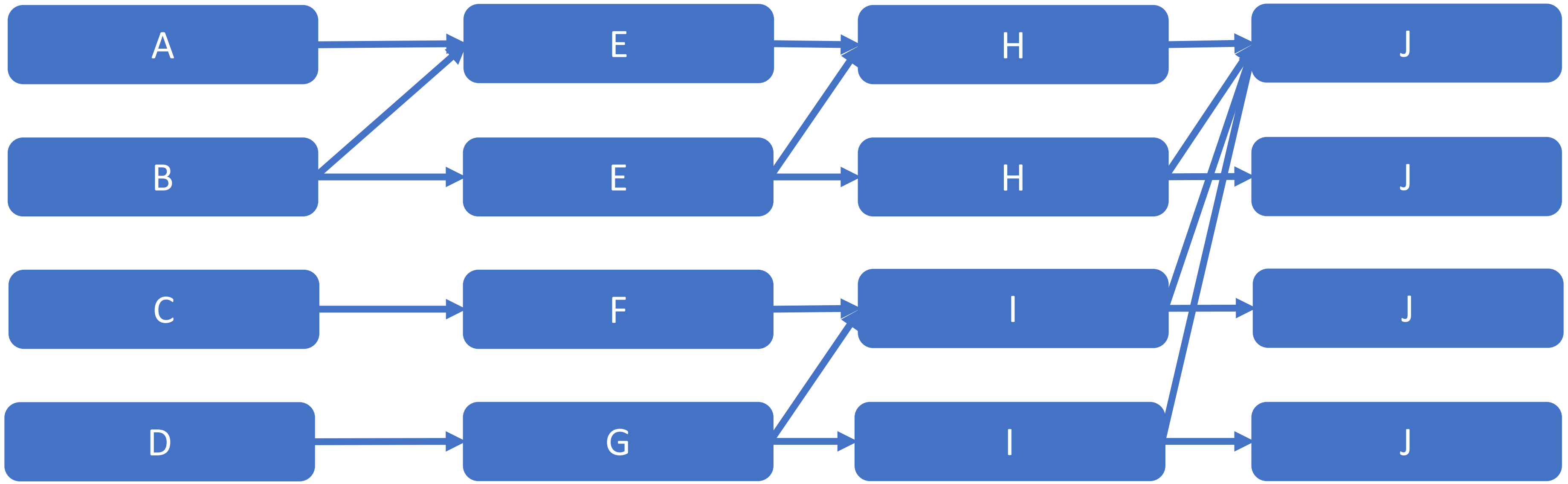
Approach: maintain every path at all times, merging as they join



Paths grow sequentially
Paths remain independent until linked
Trees form spontaneously

Methods (Tree Reconstruction)

Approach: maintain every path at all times, merging as they join



Paths grow sequentially
Paths remain independent until linked
Trees form spontaneously

Asymptotic Comparison

	APIN	AutoCRAT
I[illegible]	$O(1)$	$O(\|A\|^2)$
[illegible]	$O(\|V\| + \|A\|)$	$O(\|A\| + \|E\|^3) \subseteq O(\|A\|^3)$
[illegible]	[illegible]	$O(1)$
Retrieve Trees	[illegible]	[illegible]
Reinsert	[illegible]	[illegible]
Database Size	$O(\|V\| + \|A\|)$	[illegible]

A – alerts
V – vertices (computers)
[illegible] endpoints

APIN dominates insertion and storage; AutoCRAT conditionally dominates retrieval

*APIN ranks nodes, while AutoCRAT ranks endpoints and paths.

UTSA

Results Comparison

	APIN	AutoCRAT	APIN-Internal	AutoCRAT-Internal
[illegible]	[illegible]	13h42m41s	9s	35s
[illegible]	[illegible]	[illegible]	0.28s	5s
Retri[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Retrieve Top 20 Trees	[illegible]	[illegible]	[illegible]	[illegible]
Coverage (Nodes)	99.6%	100%	[illegible]	[illegible]
Coverage (Events	3.4%	100%	0.6%	[illegible]
Database Size	637 MB	1.1 GB	2.9 MB	2.4 MB

AutoCRAT dominates path retrieval and under comparable [illegible]

The vast majority of alerts span network borders

*APIN ranks nodes, while AutoCRAT ranks endpoints and paths.
†These ranks are inferred from their ends (for paths) or root (for trees)

UTSA

Dissertation Outline

- Introduction
- APIN: Alert Path Identification in Computer Networks
- AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees
- Alert Tree Reduction and Visualization
- Conclusion

UTSA®

Alert Tree
Reduction and Visualization

Chapter 4

Motivation

Facilitate cyber triage by selectively pruning alert trees

- Reduce visual strain
  - “Which nodes can be removed to facilitate tree interpretation?”
- Preserve salient information
  - “What nodes must be kept based on relevant metrics?”

UTSA

Motivating Example

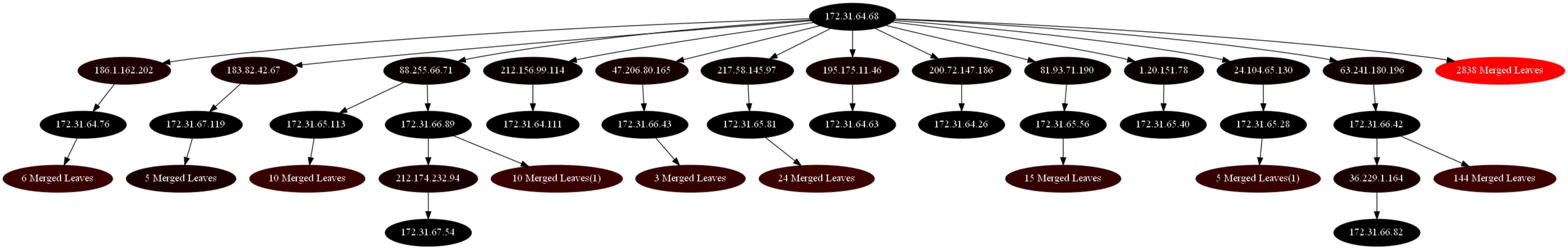
- This tree (from real data) has **3090 nodes**.
- Graphviz is forced to render it at **6% of its original resolution**.*

***If you have difficulty reading any of the node labels, that's exactly the problem we need to solve**

UTSA

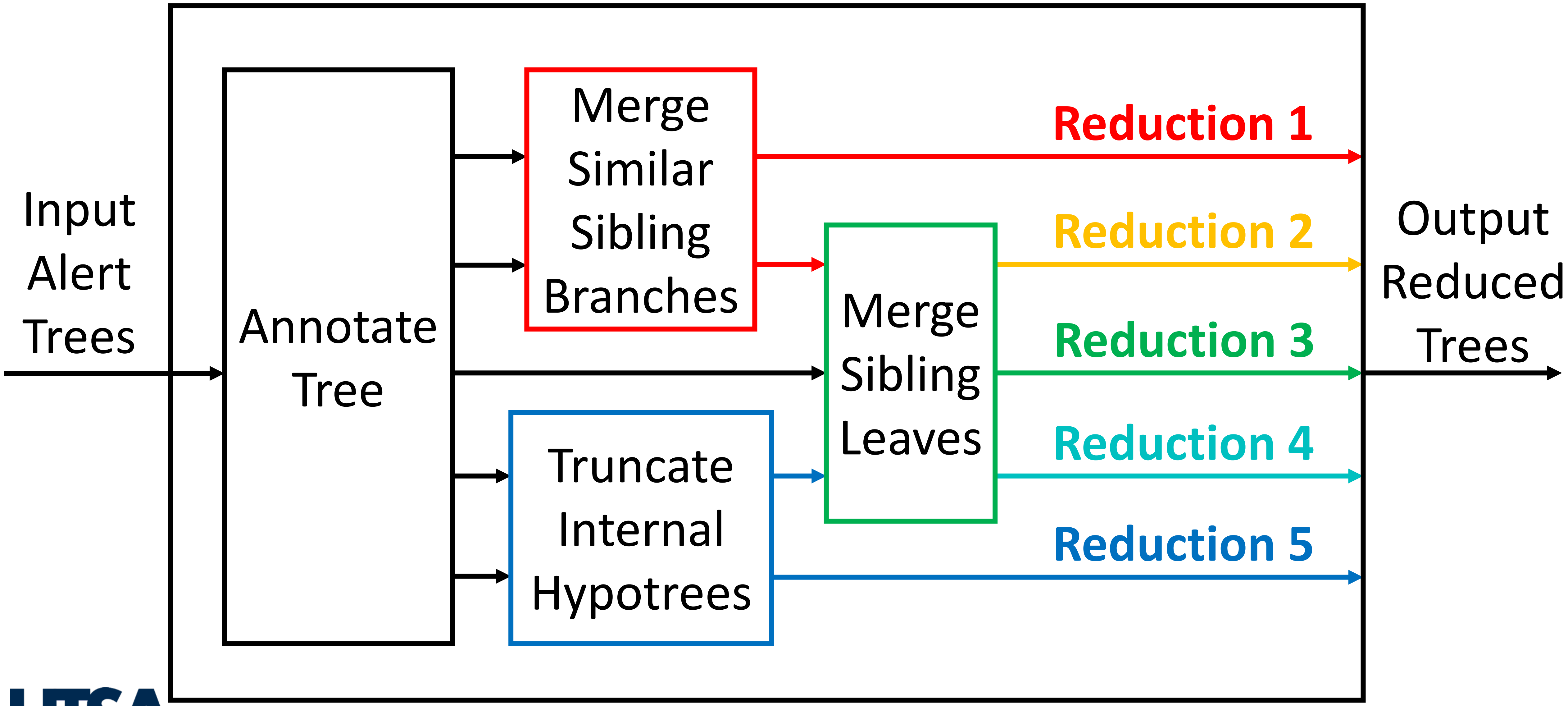
Motivating Example

- After reduction, 3090 nodes becomes 40 nodes **(98.7% reduction)**



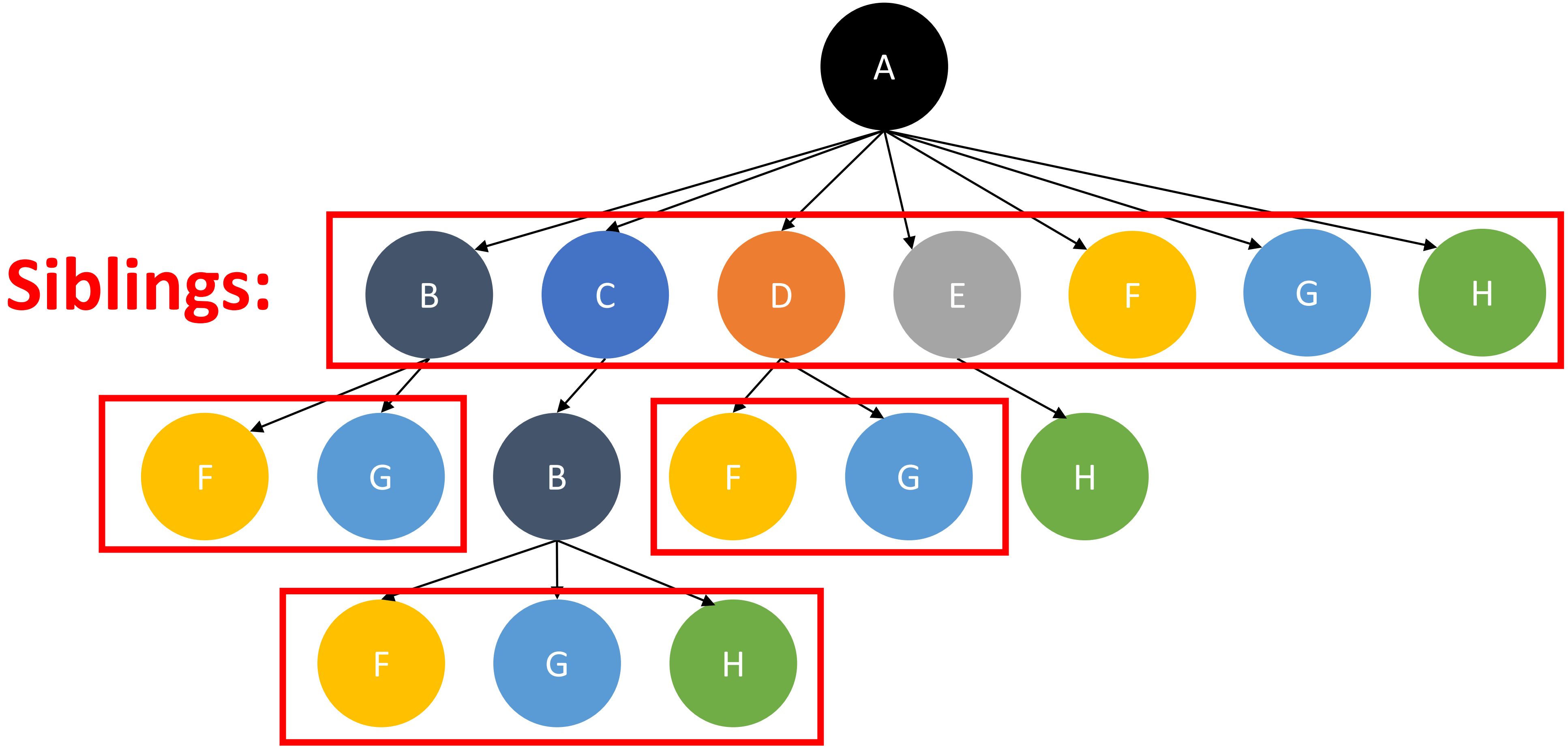
So how do we do it???

Alert Tree Reduction Architecture



UTSA

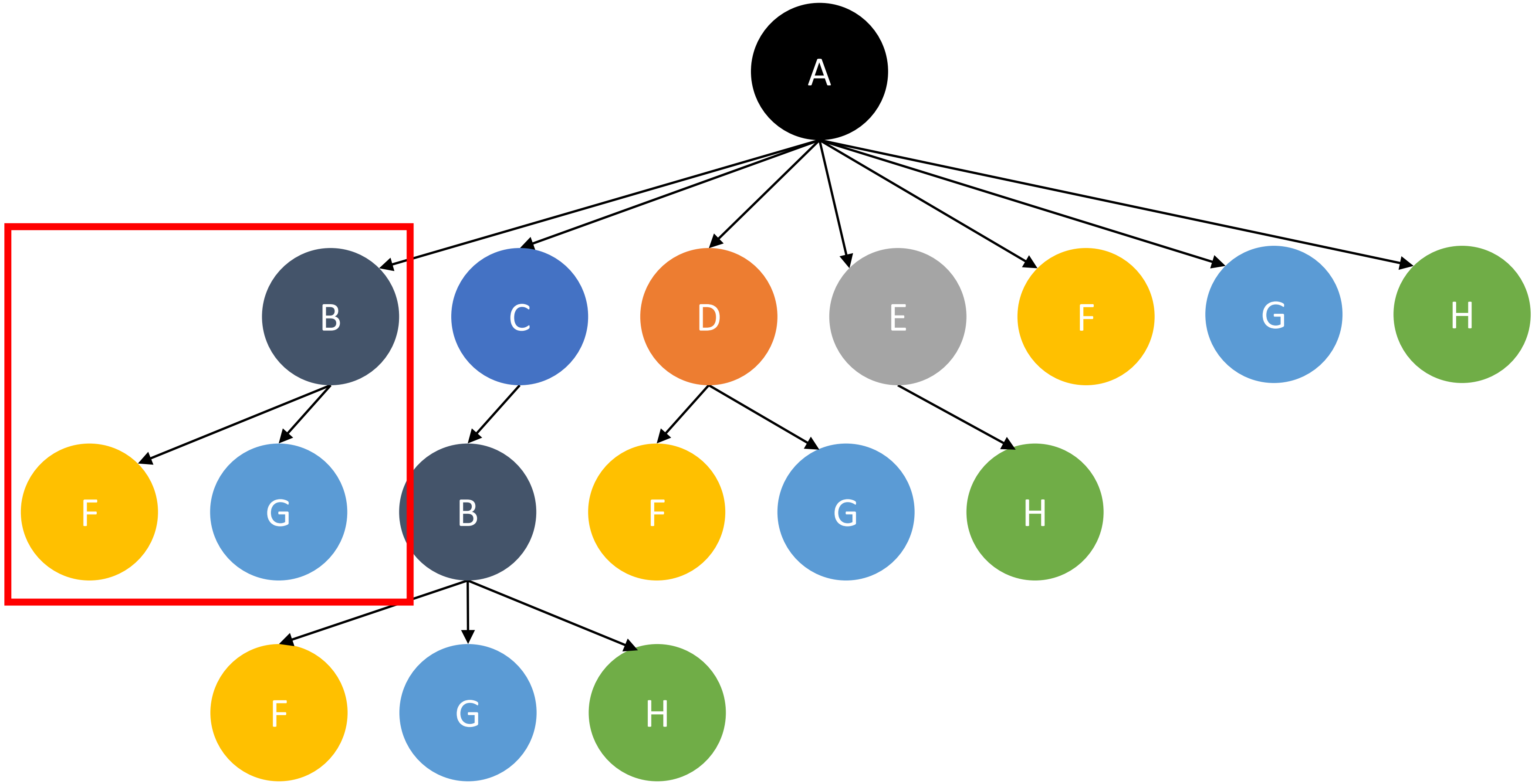
Terminology



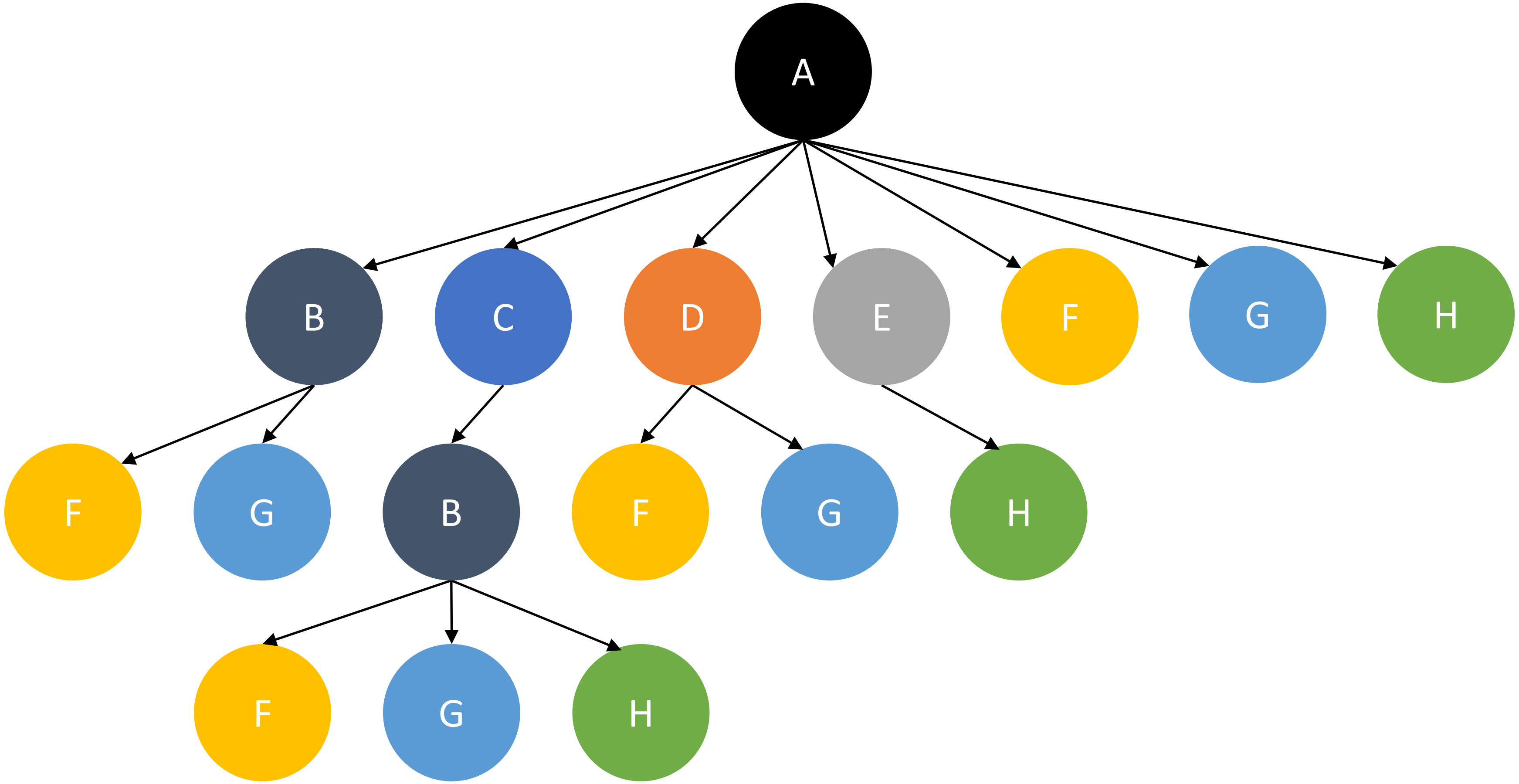
Duplicate labels may exist in a tree but not in a path

Terminology

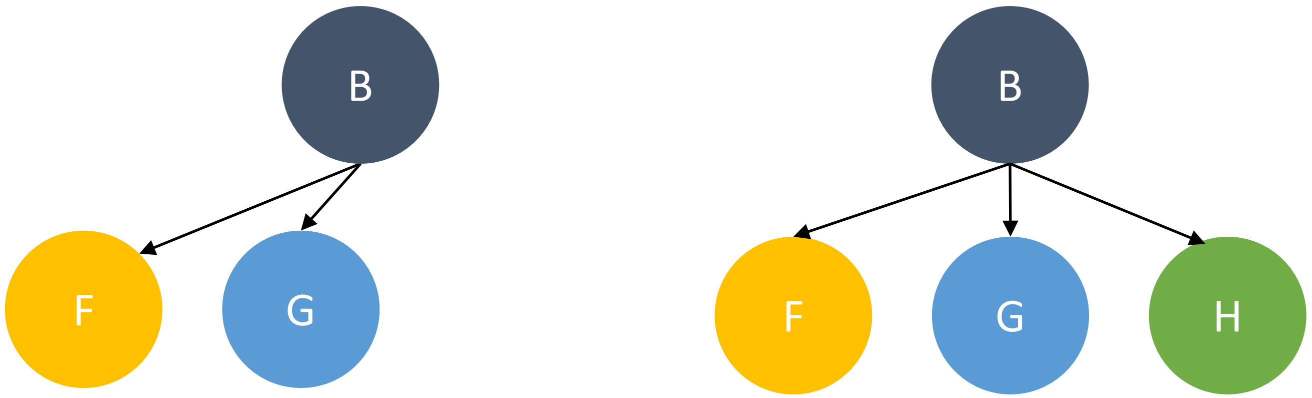
Branch:



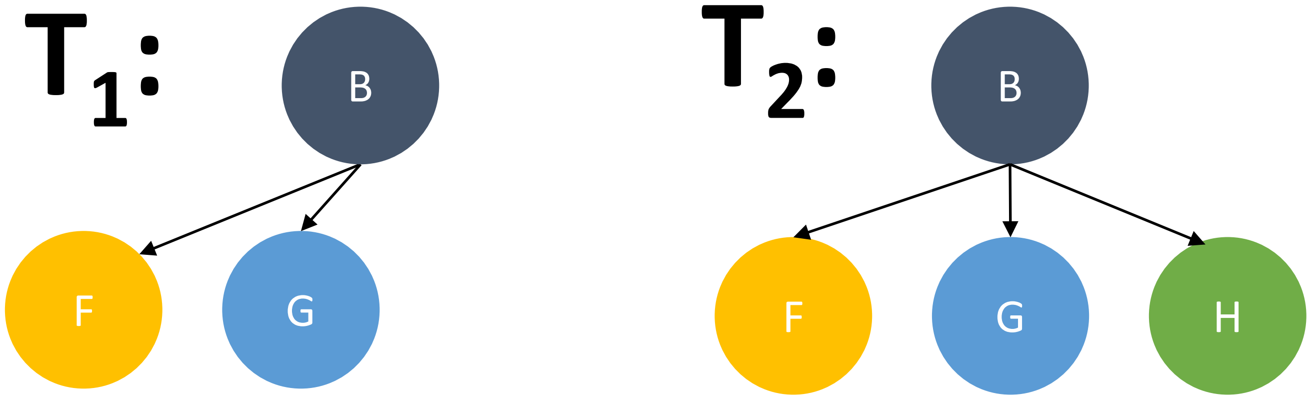
Terminology (Graph Theory vs Data Structures)



Terminology (Graph Theory vs Data Structures)



Terminology (Graph Theory vs Data Structures)



In Graph Theory:

- T_1 is a subtree of T_2
- F is a subtree of T_1 (or T_2)

In Data Structures:

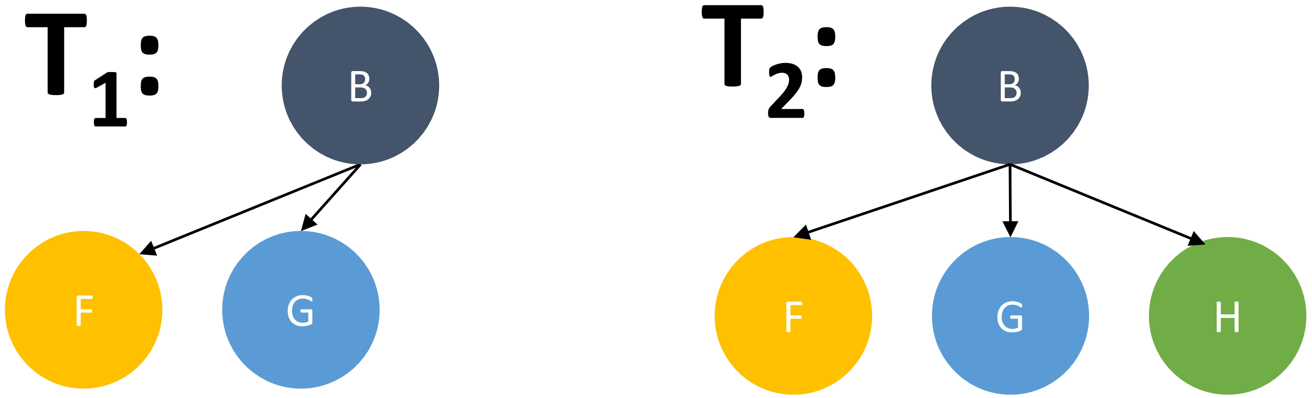
- F is a subtree of B (in either tree)

We need a new term for the relationship T_1:T_2 that eliminates ambiguity

UTSA

45

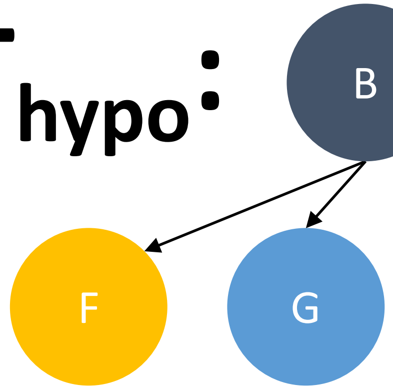
Hypotree and Hypertree



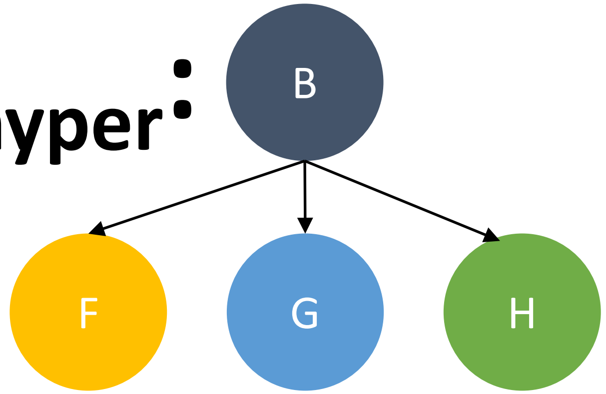
- Designate: T_1 is a hypotree of T_2 ($T_1 \trianglelefteq T_2$)
- Designate: T_2 is a hypertree of T_1 ($T_2 \trianglerighteq T_1$)
- Every tree is both a hypotree and a hypertree of itself
- We also designate proper hypotree ($\lhd$) and proper hypertree ($\rhd$)

Hypotree and Hypertree

T_{hypo}:



T_{hyper}:



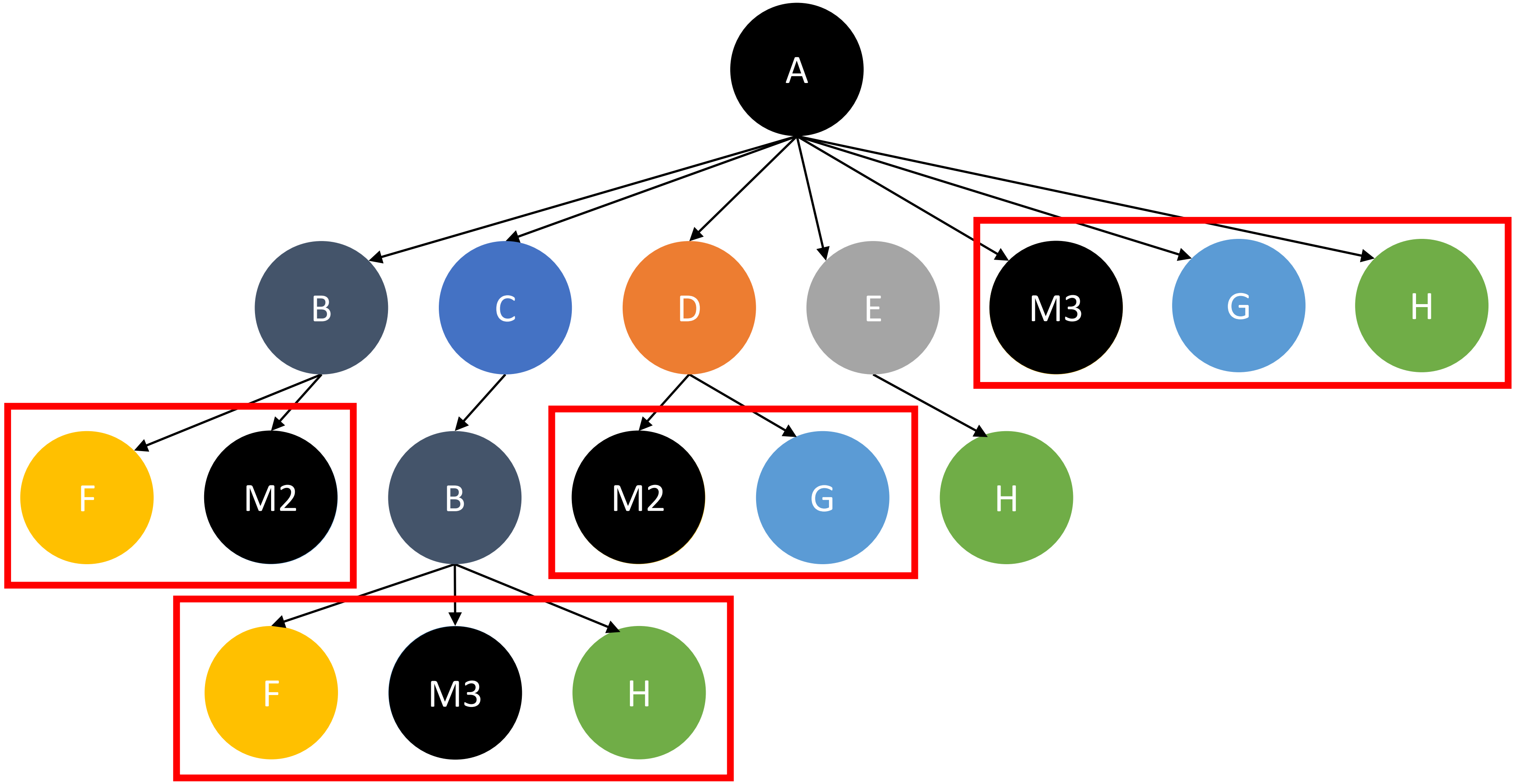
Definition: A tree T_{hypo} is a hypotree of a tree T_{hyper} if:

$\forall\ n \in T_{hypo}, \exists\ n' \in T_{hyper}$:

$\forall\ i \in \{0,1, \ldots, |n.\text{ancestors}|\},\ n.\text{ancestors}_i = n'.\text{ancestors}_i$

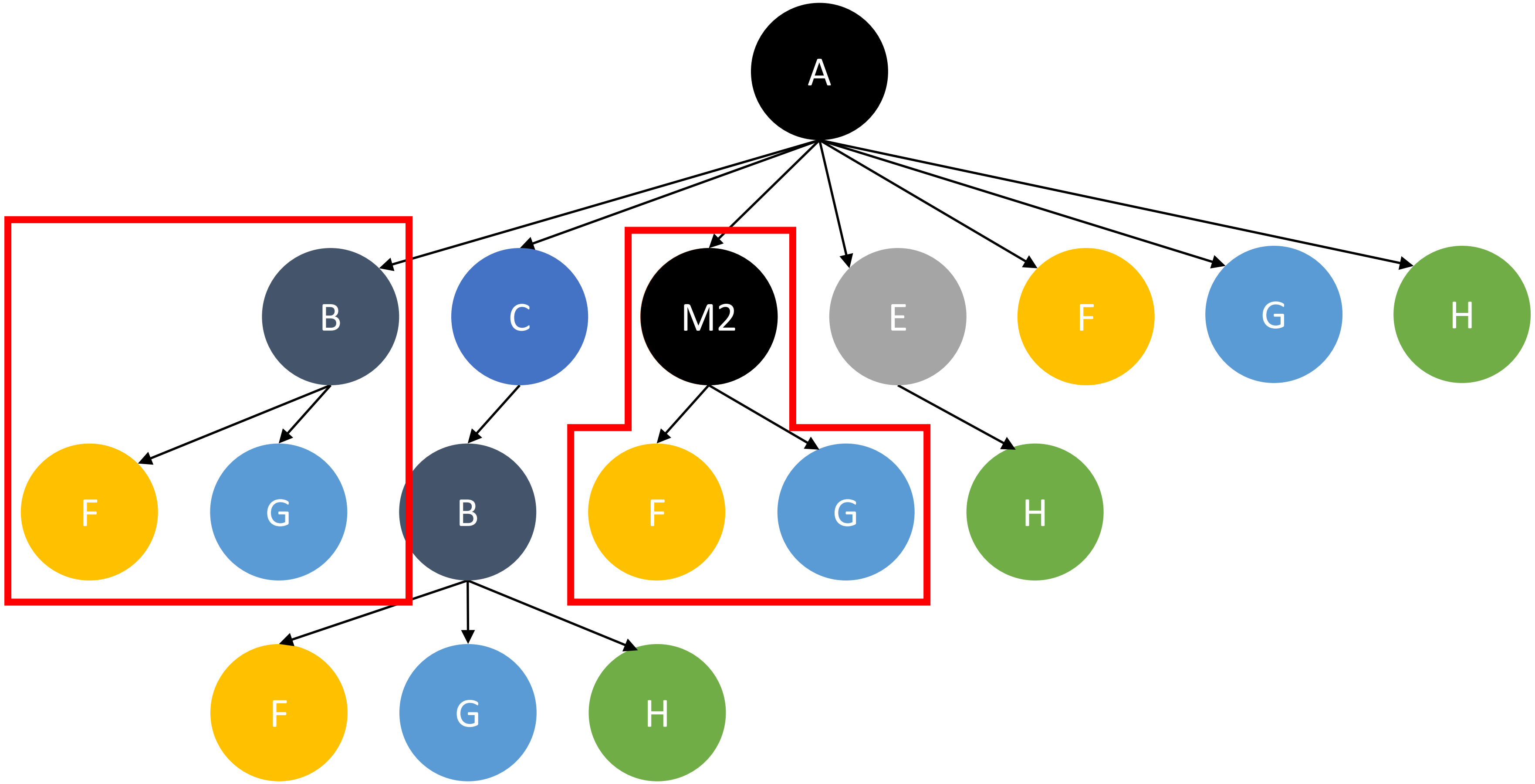
*Hypertree is derived from hypotree. Refer to the paper for exact detail

Merging Sibling Leaves [MSL(A)]



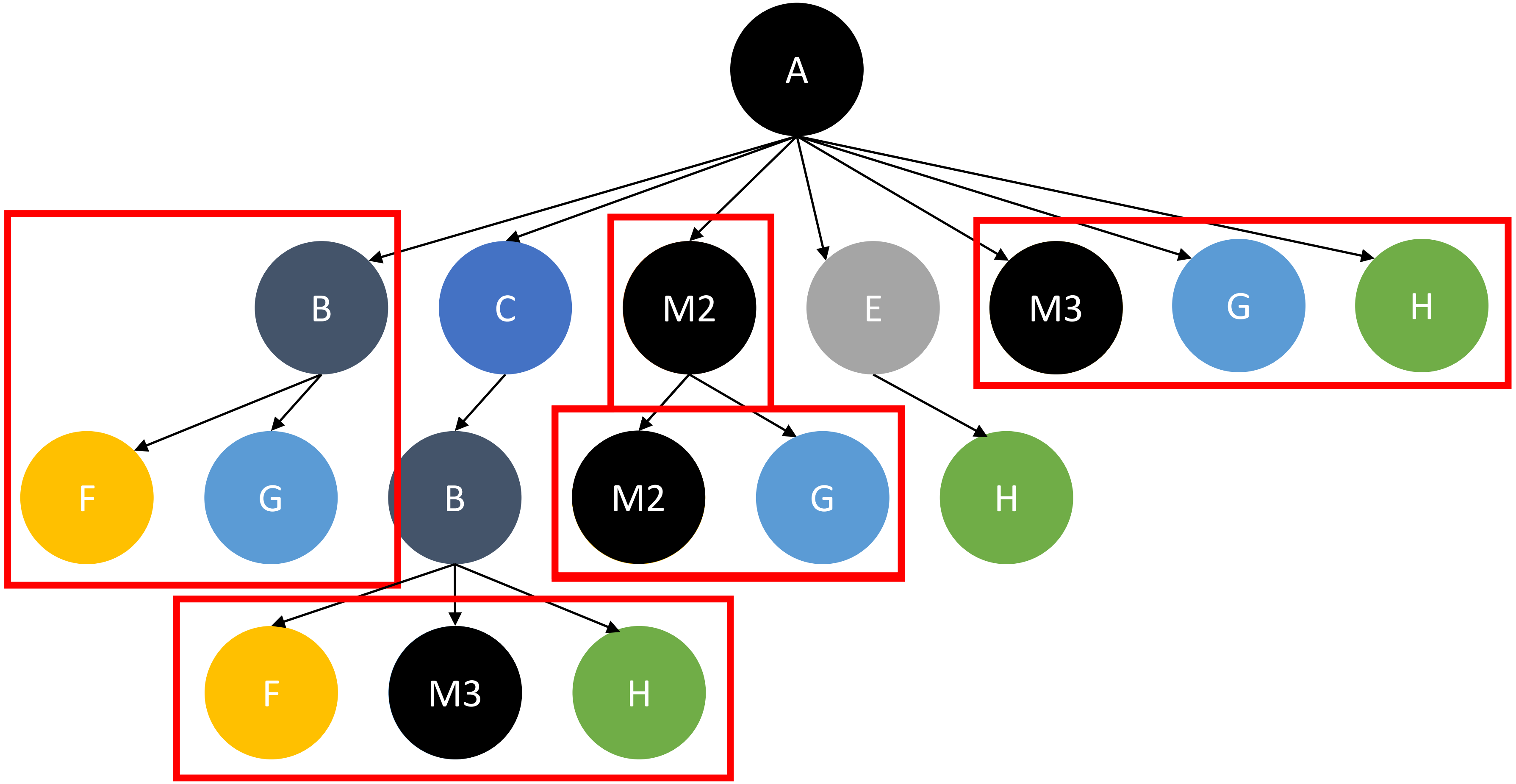
Node labels represent IP addresses
Duplicate labels may exist in a tree but not in a path within that tree
Here, node colors show labels (rather than threat score) for ease of understanding

Merging Similar Sibling Branches [MSB(A)]

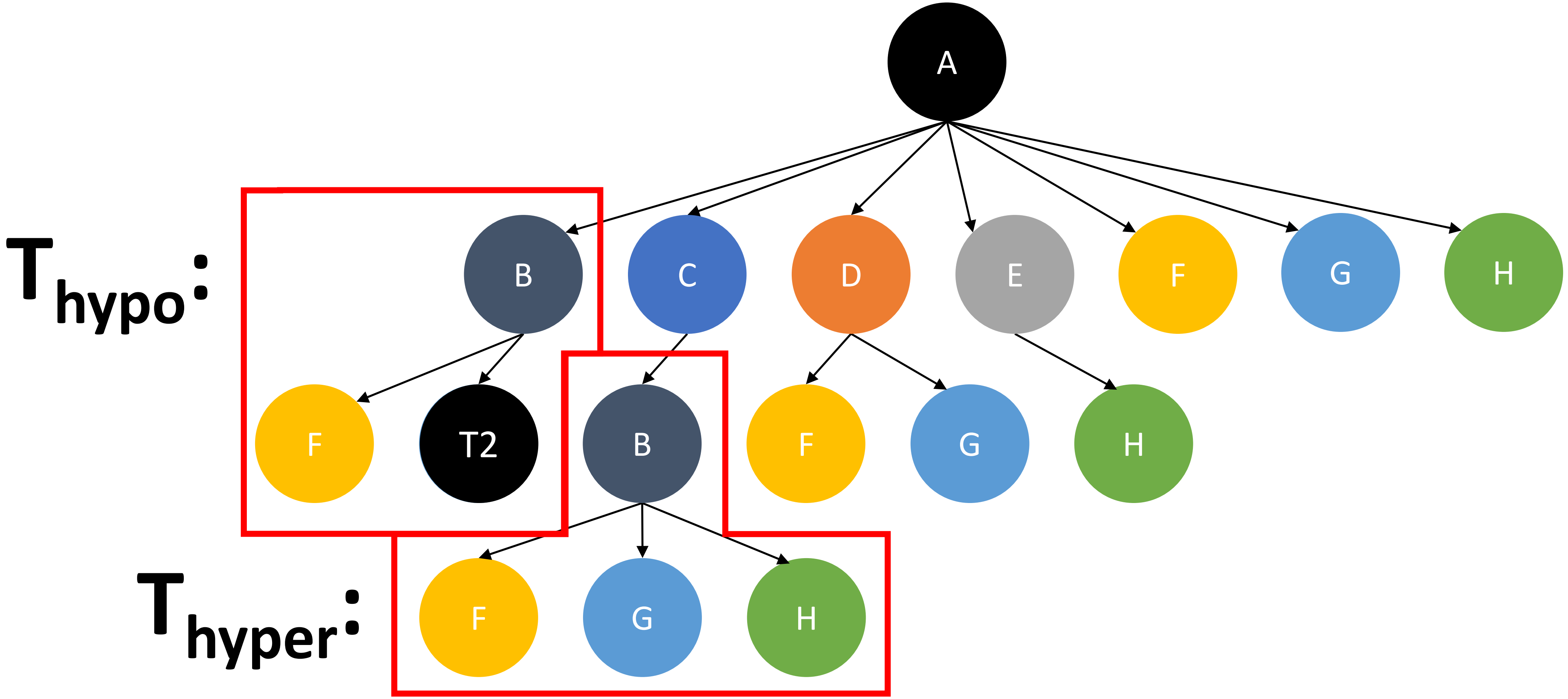


Similar Branches: A set of branches for which all subtrees excluding the branch root exist in both branches

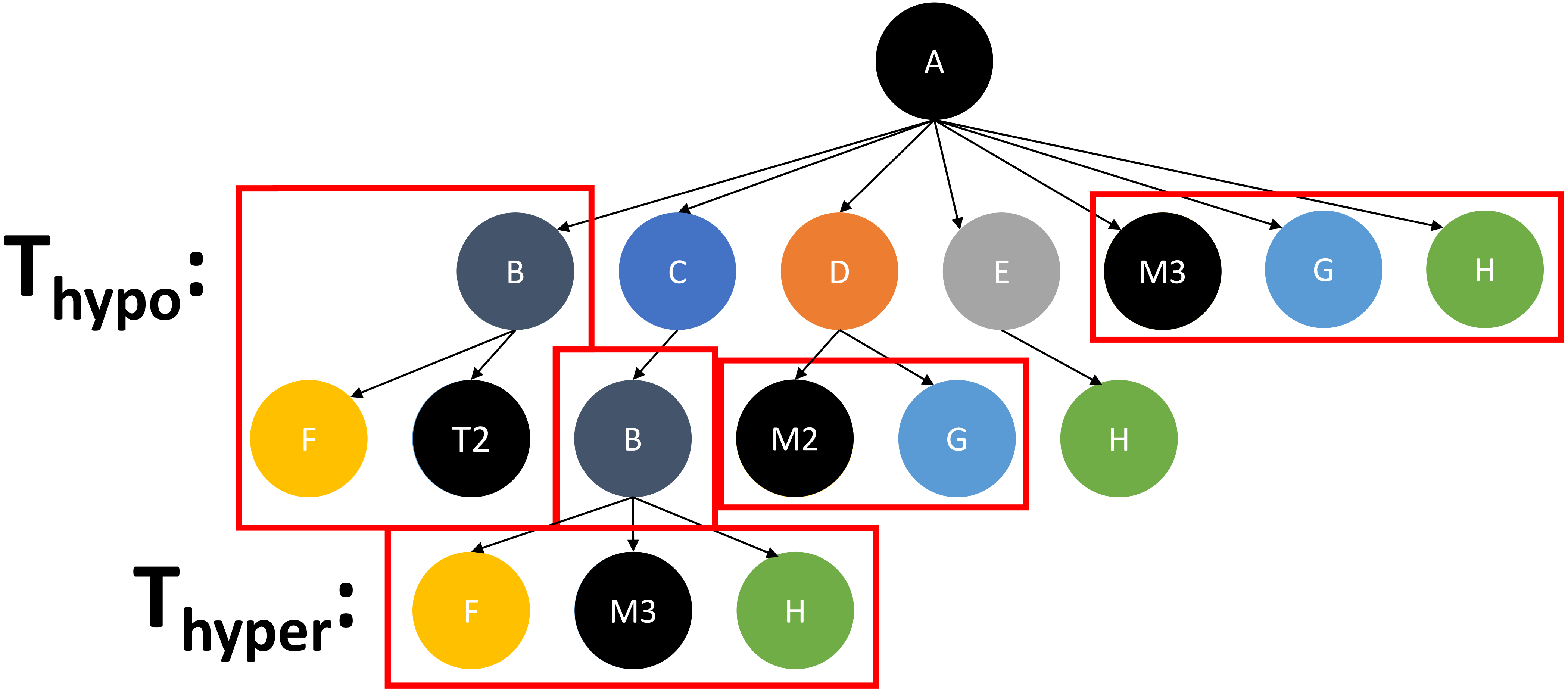
Merging Sibling Branches & Leaves [MSL(MSB(A))]



Truncating Hypotrees [TH(A)]



Truncating Hypotrees & Merging Sibling Leaves [MSL(TH(A))]



UTSA

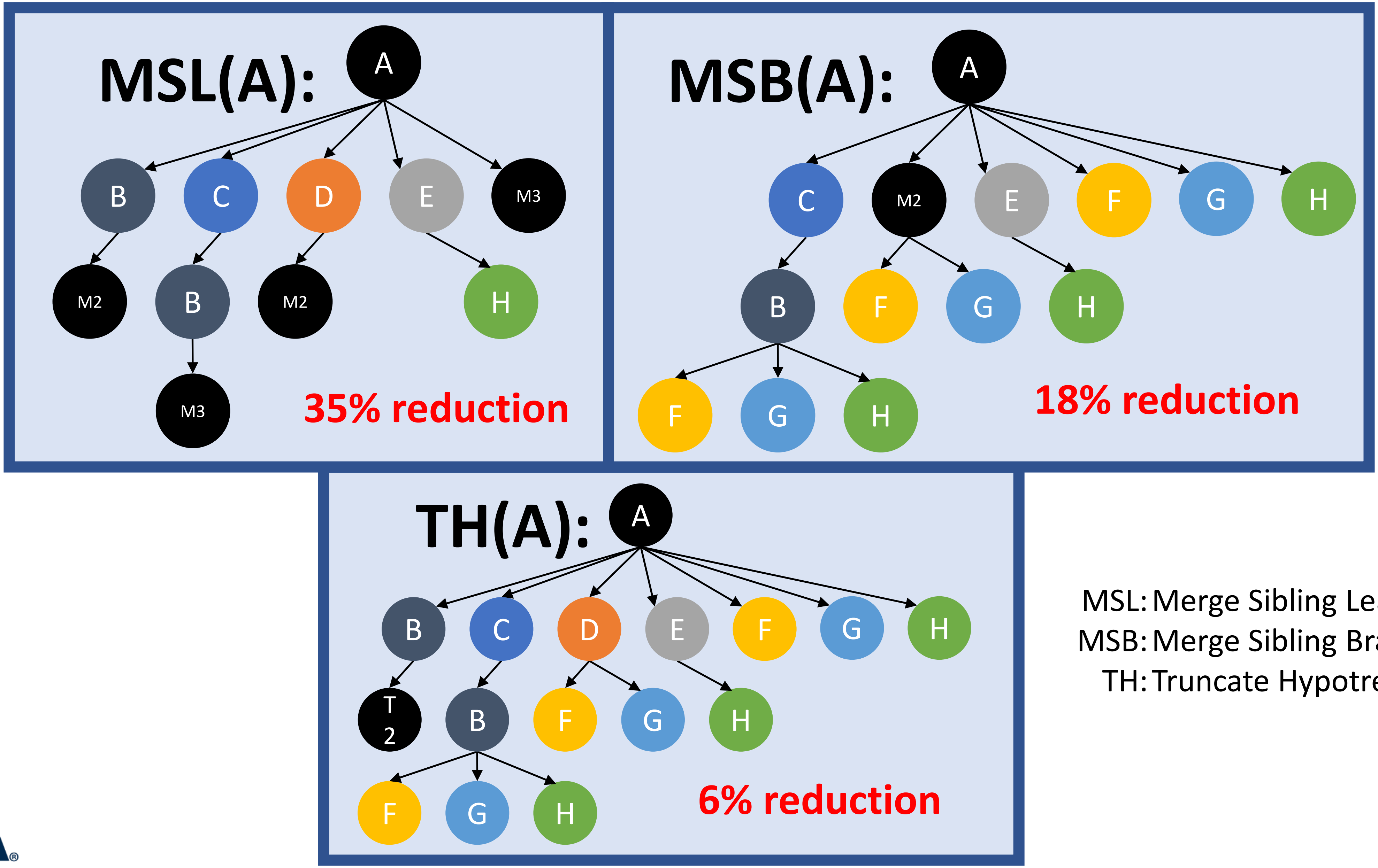
Method Restrictions

- MSL makes some trees similar (because “M2” = “M2”)
  - MSB(MSL(T)) is unsafe (but MSL(MSB(T)) is safe)
  - TH(MSL(T)) is unsafe (but MSL(TH(T)) is safe)
- MSB and TH may target the same branches
  - MSB(TH(T)) ≠ TH(MSB(T))
- The 5 valid reduction schedules:
  1. MSB(T)
  2. MSL(MSB(T))
  3. MSL(T)
  4. MSL(TH(T))
  5. TH(T)

MSL: Merge Sibling Leaves
MSB: Merge Sibling Branches
TH: Truncate Hypotrees

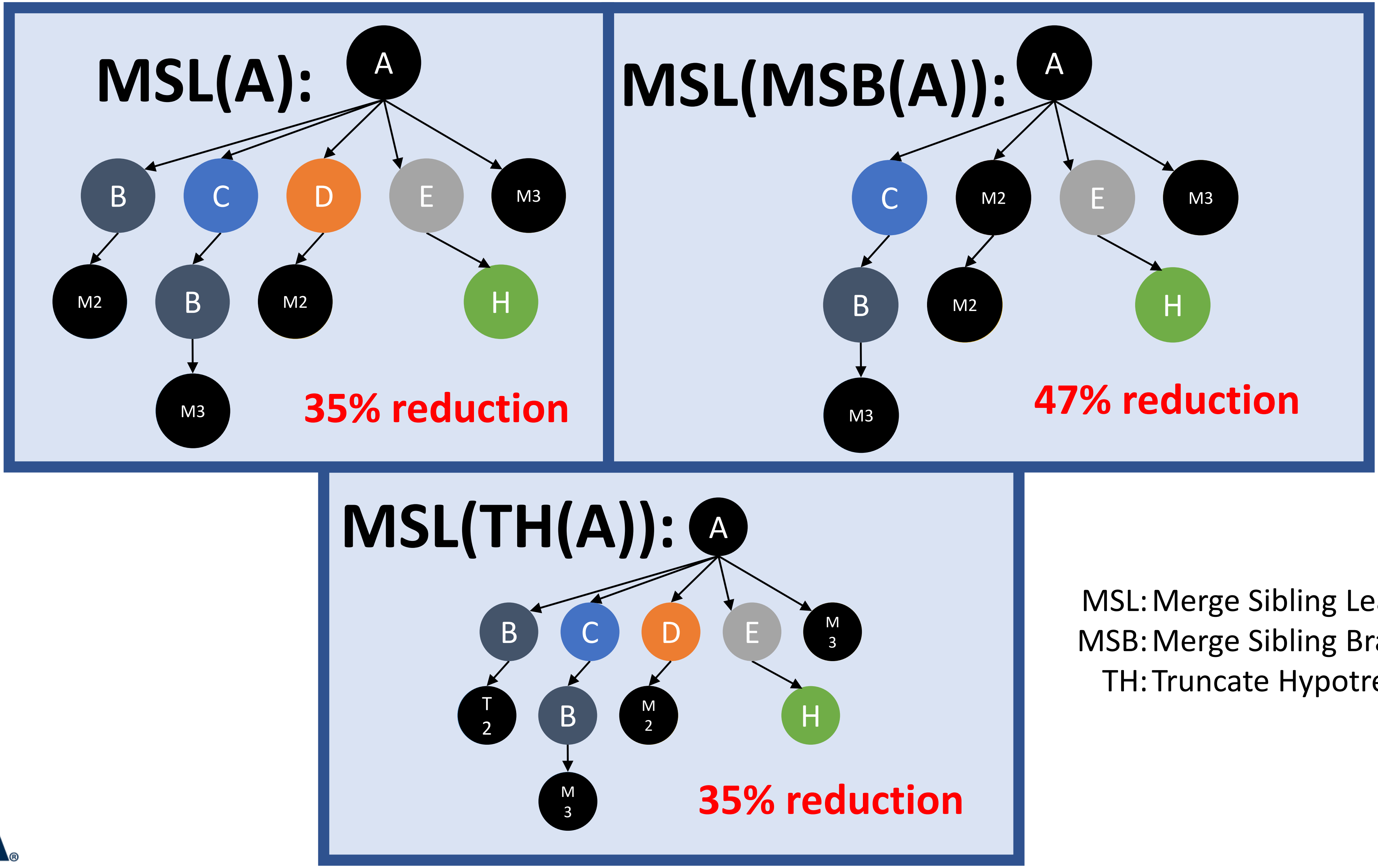
UTSA

Method Comparisons (Toy Example)



MSL: Merge Sibling Leaves
MSB: Merge Sibling Branches
TH: Truncate Hypotrees

Method Comparisons (Toy Example)



MSL: Merge Sibling Leaves
MSB: Merge Sibling Branches
TH: Truncate Hypotrees

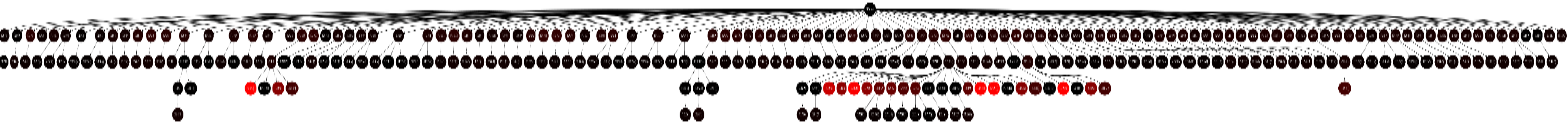
Visualization

- Black (low threat) -> red (high threat)
  - Min-max normalized
- Merged nodes
  - Color shows highest threat of those merged

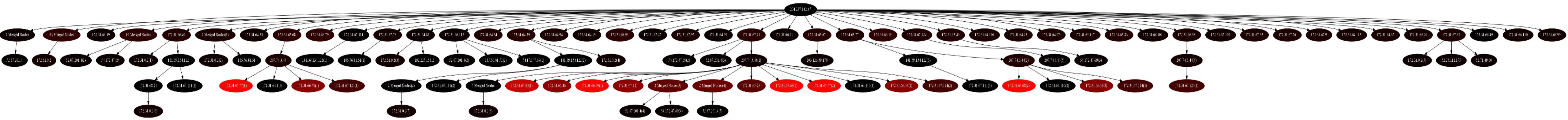
UTSA

Results (Visual): Forward Tree 204.237.142.47

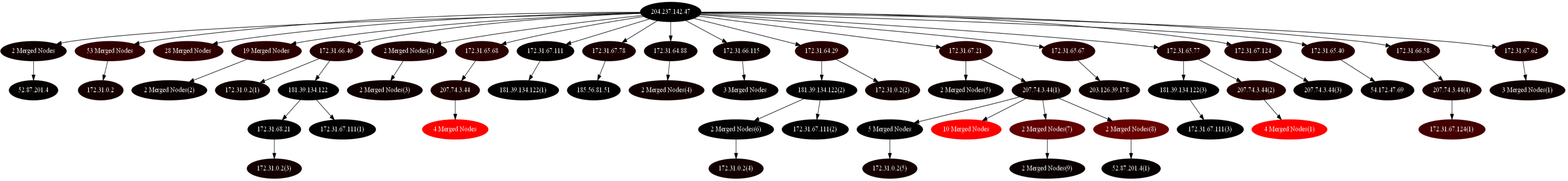
- Full Tree



- Tree with branches merged (R1)



- Tree with branches and leaves merged (R2)



Metrics

- Visual Strain Reduction (VSR)
- Node Retention (NR)
- Threat Score Retention (TSR)
- Reduction Index (RI)
  - $RI = 3/(VSR^{-1} + NR^{-1} + TSR^{-1})$

Results (Numerical)

Reduction	Tree Set	VSR	NR	TSR	RI
MSB	Top 5	0.243	0.539	0.278	0.313
	Random 5	0.352	0.553	0.254	0.349
	Bottom 5	0.433	0.493	0.36	0.42
MSL	Top 5	0.363	0.577	0.611	0.489
	Random 5	0.282	0.824	0.799	0.499
	Bottom 5	0.791	0.744	0.73	0.754
TH	Top 5	0.009	1	0.999	0.026
	Random 5	0	1	1	0
	Bottom 5	0.037	1	0.983	0.103

MSB: Merge Sibling Branches
MSL: Merge Sibling Leaves
TH: Truncate Hypotrees

VSR: Visual Strain Reduction
NR: Node Retention
TSR: Threat Score Retention
RI: Reduction Index

UTSA®

Dissertation Outline

- Introduction
- APIN: Alert Path Identification in Computer Networks
- AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees
- Alert Tree Reduction and Visualization
- Conclusion

Conclusion

Chapter 5

Discussion

- APIN
  - Relies on network segmentation
  - Dominates maintenance time
- AutoCRAT
  - Relies on ordering assumption
  - Dominates retrieval time
- Reduction improves visualization

eric.ficke@utsa.edu

Images courtesy of pixabay.com and publicdomainvectors.org

UTSA