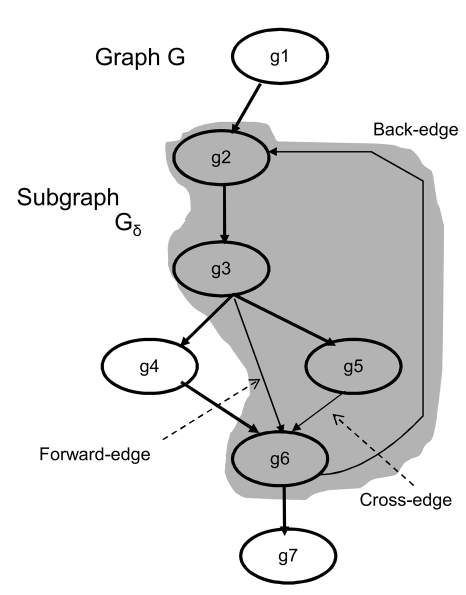
CERIAS

the center for education and research in information assurance and security

Integrity of Graphs Without Leaking

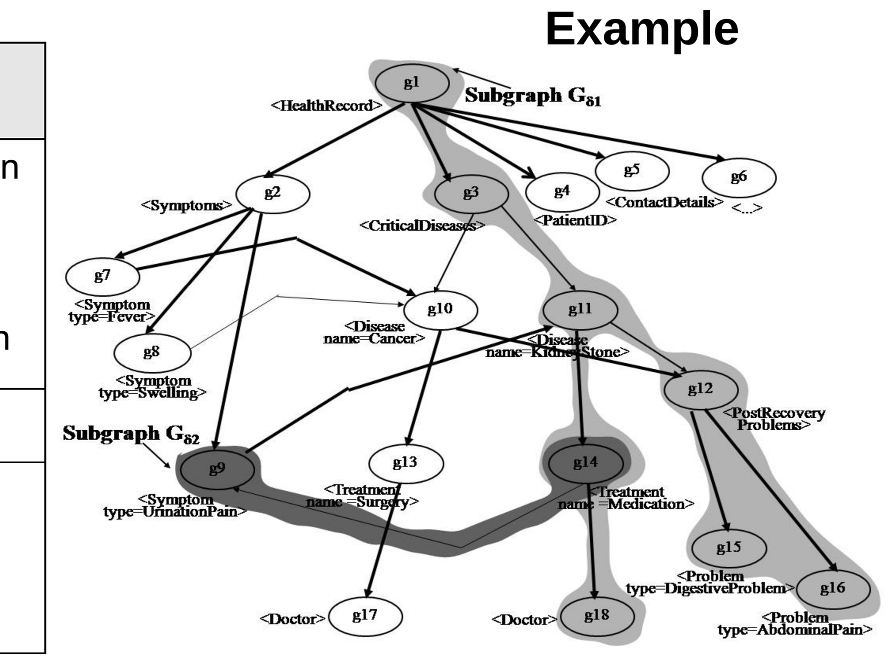
Ashish Kundu, Elisa Bertino | CS & CERIAS, Purdue University | {ashishk, bertino}@cs.purdue.edu



Applications	• Secure data distribution: Biological, Military. • Cloud computing, Trusted systems.
Prior art	• No existing prior solution for cyclic graphs. • Solution for DAGs leaks [Martel *et al*].
Challenges	• Strong security requirement. • Graphs are complex (much more than trees).
Our solution	• **Provably (Cryptographically) Secure** • **DAGs: Optimal cost** • **Graphs with cycles: Optimal cost**

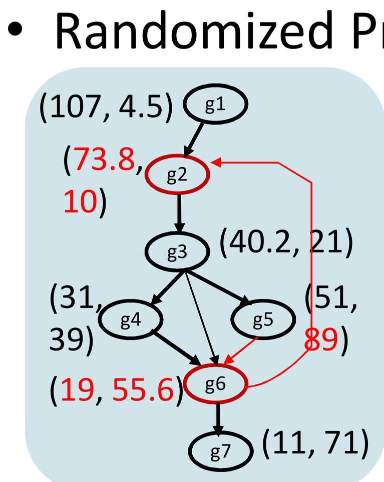
Edge e(w,y)	***Leakages***
Forward-edge	1. In-degree(y) ≥ 2, No. of edges incident on y ≥ 2 2. One edge e' is a tree-edge 3. One more node x: w…x…y is a path 4. Source graph is larger than the subgraph
Cross-edge	• 1, 2, & 4.
Back-edge	1. At least one path from y to w 2. At least one cycle in the graph 3. Cycle is between w and y. & 4.

Healthcare Example



Role of Traversal Numbers

- DAG = {DFT, Forward-edges, Cross-edges}
- Cyclic Graph = {DAG, Back-edges}
- Randomized Post-order Numbers (RPONs)
- Randomized Pre-order Numbers (RRONs)



(RPON, RRON)

Lemma 1 *Let τ be the depth-first tree of a graph $\mathcal{G}(V, E)$. Let x, $y \in V$, and $e(x, y) \in E$. Let o_x and q_x refer to PON and RON of node x, respectively. With respect to the DFT τ, $e(x, y)$ is a*

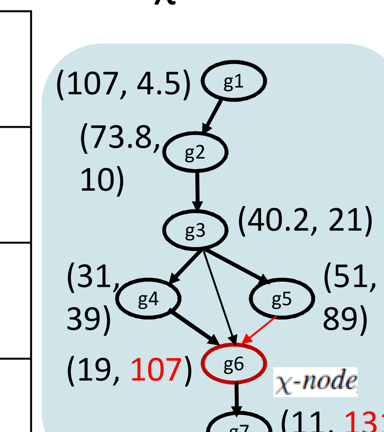
- *tree-edge iff $o_x > o_y$, and $q_x < q_y$.*
- *forward-edge iff $o_x > o_y$, and $q_x < q_y$.*
- *cross-edge iff $o_x > o_y$, and $q_x > q_y$.*
- *back-edge iff $o_x < o_y$, and $q_x > q_y$.*

Convey every edge as a **Tree-edge** (τ-edge)

DAGs: χ-RRONs

- χ-node 'x': endpoint of cross-edge(s).
  - Every other node is a τ-node.
- For each χ-node 'x' and each 'y' reachable from 'x':
  - Compute χ-RRON: $r_x^\chi > r_y$, r_y = τ-RRON or χ-RRON

Θ_x: structural position of x	χ-node: $\Theta_x = (p_x^\tau, r_x^\chi)$ τ-node: $\Theta_x = (p_x^\tau, r_x^\tau)$
Ψ_G : signature of G	$\Psi_G = H(\Theta_1, \ldots, \Theta_n)$; Sign Ψ_G.
Ψ_x: signature of x	$\Psi_x = H(\Psi_G, \Theta_x, C_x)$; Sign Ψ_x.
Ψ_δ : signature of the set of nodes V_δ in subgraph G_δ	$x \in V_\delta$: Ψ_δ = Aggregate Signature of Ψ_x



(RPON, τ-RRON/ χ-RRON)

DAGs: Verification

Ψ_δ, $\{\Theta_x \mid x \in V_\delta\}$, G_δ

Distributor → User

(1) Verify	Compute Aggregate Signature Ψ_δ'. $\Psi_\delta' \neq$ received Ψ_δ: G_δ compromised.
(2) Edge e(z, x)	$(p_x \geq p_z)$ OR $(r_x \geq r_z)$: e(z, x) is compromised.
(3) Sibling order (x, y)	$(p_x \geq p_y)$ OR $(r_x \geq r_y)$: (x, y) order compromised.
(4) Content	Verification (1) Fails: Content C_x or Θ_x compromised.

✓ **No leakage**:
- every edge e(z, x) is conveyed as a **tree-edge**.

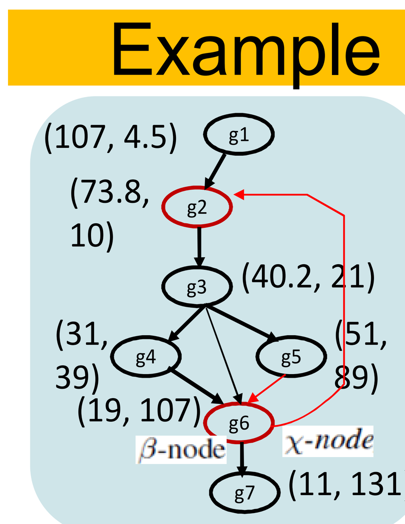
Cyclic Graphs: β-RRONs,β-RPONs

- β-node 'x': *start* node of a back-edge e(x, w). (g_6)
- β-reachable 'y': node reachable from 'x' over e(x, w). (g_2, g_3, g_4, g_5)
- For each 'y', β-reachable from 'x':
  - Compute β-RRON: $r_y^\beta > r_x^\beta$, r_x^β = τ-RRON or χ-RRON of x.
  - Compute β-RPON: $p_y^\beta < p_x^\beta$, p_x^β = τ-RPON or χ-RPON of x.

Θ_x^β: structural position of x	$\Theta_x^\beta = (p_x^\beta, r_x^\beta)$, $\Theta_x^\alpha = (p_x^\tau, r_x^\tau)$, or (p_x^τ, r_x^χ).
Ψ_G : signature of G	$\Psi_G = H(\Theta_1^{\alpha\mid\beta}, \ldots, \Theta_n^{\alpha\mid\beta})$; Sign Ψ_G.
Ψ_x^β: signature of x	$\Psi_x^\beta = H(\Psi_G, \Theta_x^\beta, C_x)$; Sign Ψ_x.

Ψ_δ : signature of the set of nodes V_δ in subgraph G_δ	$x \in V_\delta$: • $\Omega = \phi$. • If x is β-reachable or a β-node in G_δ, $\Omega = \Omega \cup \{\Theta_x^\beta\}$, Else $\Omega = \Omega \cup \{\Theta_x^\alpha\}$. • Ψ_δ = Aggregate Signature of $\Psi_x \in \Omega$.

Example



(RPON, τ-RRON/ χ-RRON)

β-reachable	(β-RPON, β-RRON)
g_2	(6, 145)
g_3	(-16, 156)
g_4	(-29, 181)
g_5	(-45, 223)

Cyclic Graphs: Verification

Ψ_δ, $\{\Theta_x \mid x \in \Omega\}$, G_δ

Distributor → User

(1) Verify	Compute Aggregate Signature Ψ_δ'. $\Psi_\delta' \neq$ received Ψ_δ: G_δ compromised.
(2) Edge e(z, x)	$(p_x \geq p_z)$ OR $(r_x \geq r_z)$: e(z, x) is compromised.
(3) Content	Verification (1) Fails: Content C_x or Θ_x compromised.

✓ **No leakage**:
- If G_δ does not have any cycle, every edge e(z, x) is conveyed as a **tree-edge**.
- Else knowledge of back-edge does **not** leak any information.

Summary

- We showed that how knowledge of edge-types can be exploited to infer sensitive information.
- First such technique for strong security for DAGs & Graphs

Provably secure, privacy-preserving	Integrity and confidentiality (leakage-free)
Efficient, Optimal	• Only **Constant (O(1))** number of signature items to be sent to the user. • DAGs: Linear (O(n)) sig. items to be computed. • Cyclic graphs: Optimal (O(n*d)) sig. items to be computed.
Simple, Easy to implement	post-order and pre-order traversals are simple to understand and implement.

Security

- Integrity: Proof by
  - reduction to security of cryptographic hash functions
  - reduction to security of aggregate signatures [Boneh *et al*].
- Confidentiality: Proof
  - Randomized traversal numbers are secure. [VLDB'08]
  - Simple: addition or sorting of random numbers

References

- **Structural Signatures for Tree Data Structures**, *Ashish Kundu & Elisa Bertino, VLDB '08.*
- **Completely-Secure Sharing of Trees and Hierarchical Content**, *Ashish Kundu & Elisa Bertino, CERIAS Symposium '07.* (Best poster: 2nd)
- **Secure Dissemination of XML Content Using Structure-based Routing**, *Ashish Kundu & Elisa Bertino, IEEE EDOC '06.* (Best student paper)
- **Structural Signatures for Graph Data Structures**, *Ashish Kundu & Elisa Bertino,, Ready for submission.*

PURDUE UNIVERSITY

CERIAS

Discovery Park
e-Enterprise Center