Probabilistic reasoning with graphical security models

Barbara Kordy

Clermont-Ferrand, January 7, 2016

Digital Confidence seminar

UMR IRISA

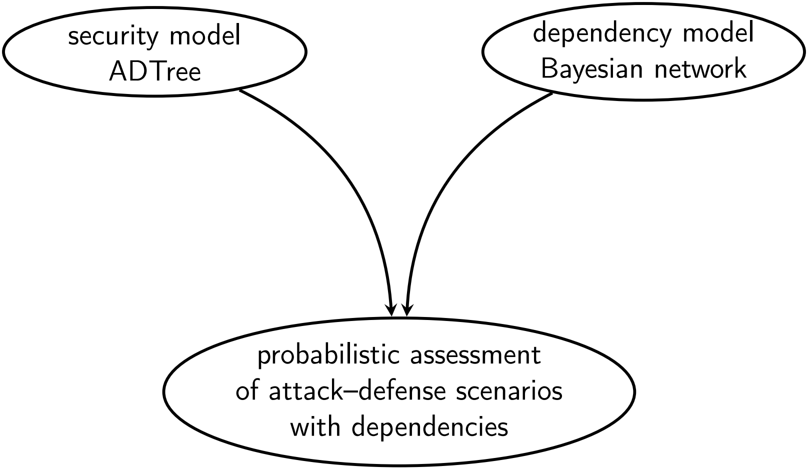
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Joint work

Prof. Dr. Marc Pouly
Lucerne University of Applied Sciences and Arts

Dr. Patrick Schweitzer
University of Luxembourg

Probabilistic assessment of security scenarios



Outline

Modeling security scenarios

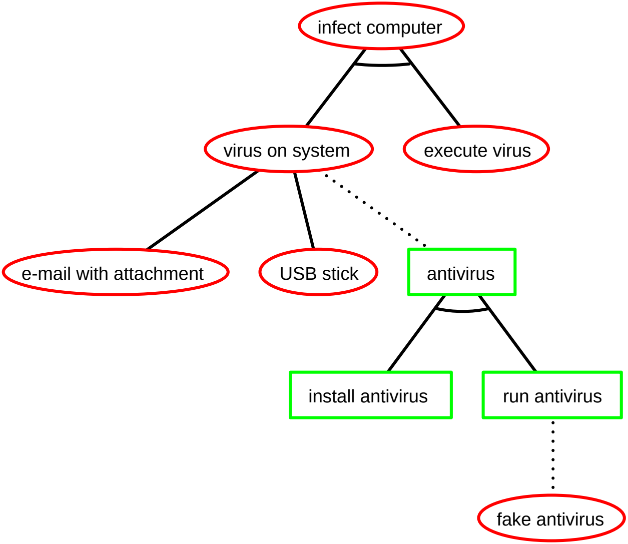
Attack–defense tree (ADTree) [JLC'14]

Tree-like representation of an attack–defense scenario depicting:

- How to attack a system
- How to protect against an attack

- Extend the **industrially recognized** model of attack trees [Schneier'99]
- Integrate
  - **Intuitive** representation features [IJSSE'12, ICISC'12]
  - **Formal** analysis techniques [GameSec'10, SIIS'11, JLC'14]
  - **Software** application ADTool [QEST'13]

Example: ADTree for infecting a computer



Propositional semantics for ADTrees [SIIS'11]

$\mathcal{B}$ – the set of non-refined nodes of ADTree t

- $\mathbf{x} \in \{0,1\}^{\mathcal{B}}$ encodes whether actions from $\mathcal{B}$ succeed or not
  - Action $A \in \mathcal{B}$ succeeds if $\mathbf{x}(A) = 1$
  - Action $A \in \mathcal{B}$ does not succeed if $\mathbf{x}(A) = 0$

Boolean function f_t for t

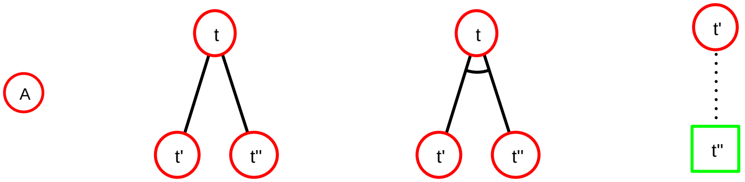
$f_t \colon \{0,1\}^{\mathcal{B}} \to \{0,1\}$ associates a Boolean value $f_t(\mathbf{x}) \in \{0,1\}$ with each vector $\mathbf{x} \in \{0,1\}^{\mathcal{B}}$

$\mathbf{x}$ is called an **attack vector** if $f_t(\mathbf{x}) = 1$

ADTrees as Boolean functions

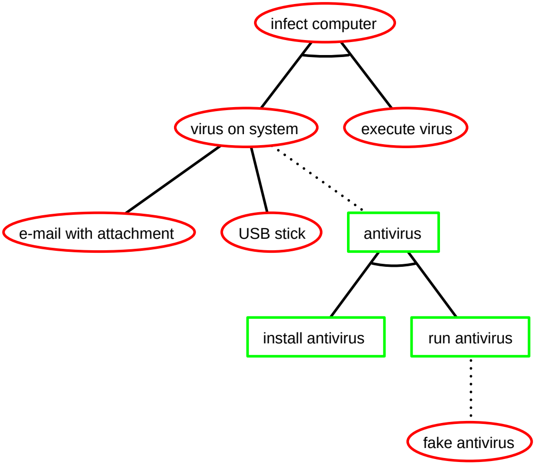
Domain of f_t is composed of the non-refined nodes of t

Non-refined | OR | AND | Countermeasure



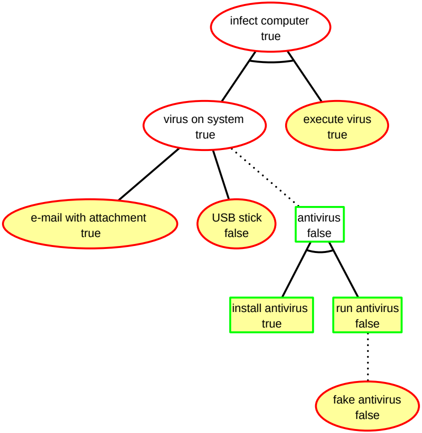
$f_t(A) = A$ | $f_t = f_{t'} \vee f_{t''}$ | $f_t = f_{t'} \wedge f_{t''}$ | $f_t = f_{t'} \wedge \neg f_{t''}$

Example: Boolean function for infecting a computer



$$f_t = \Big(\big(X_{\mathrm{EA}} \vee X_{\mathrm{US}}\big) \wedge \neg\big(X_{\mathrm{IA}} \wedge (X_{\mathrm{RA}} \wedge \neg X_{\mathrm{FA}})\big)\Big) \wedge X_{\mathrm{EV}}$$

Example: attack vector



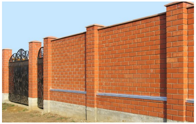
$$f_t = \Big((X_{\text{EA}} \vee X_{\text{US}}) \wedge \neg(X_{\text{IA}} \wedge (X_{\text{RA}} \wedge \neg X_{\text{FA}}))\Big) \wedge X_{\text{EV}}$$

attack vector 1 0 1 0 0 1

Importance of probabilities

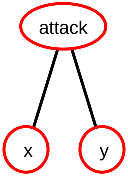
Knowing the **probabilities** of particular attacks allow us to

- Identify **the most vulnerable components**
- Determine **the strategic points**
- Decide **which protective measures to implement**

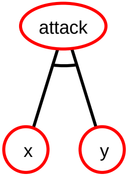


Bottom-up evaluation of probability on ADTrees [ICISC'12]

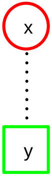
Probability of a disjunctive subtree



Probability of a conjunctive subtree

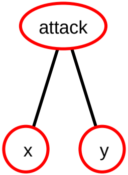


Probability of a countered subtree



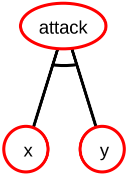
Bottom-up evaluation of probability on ADTrees [ICISC'12]

Probability of a disjunctive subtree

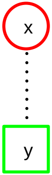


$x + y - xy$

Probability of a conjunctive subtree

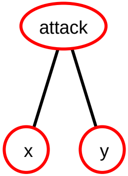


Probability of a countered subtree



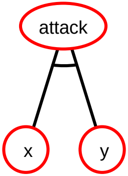
Bottom-up evaluation of probability on ADTrees [ICISC'12]

Probability of a disjunctive subtree



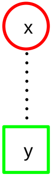
$x + y - xy$

Probability of a conjunctive subtree



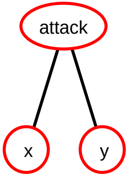
xy

Probability of a countered subtree



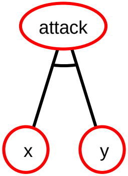
Bottom-up evaluation of probability on ADTrees [ICISC'12]

Probability of a disjunctive subtree



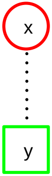
$x + y - xy$

Probability of a conjunctive subtree



xy

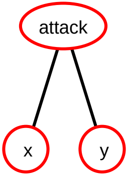
Probability of a countered subtree



$x(1 - y)$

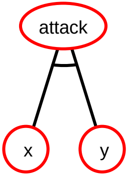
Bottom-up evaluation of probability on ADTrees [ICISC'12]

Probability of a disjunctive subtree



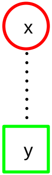
$x + y - xy$

Probability of a conjunctive subtree



xy

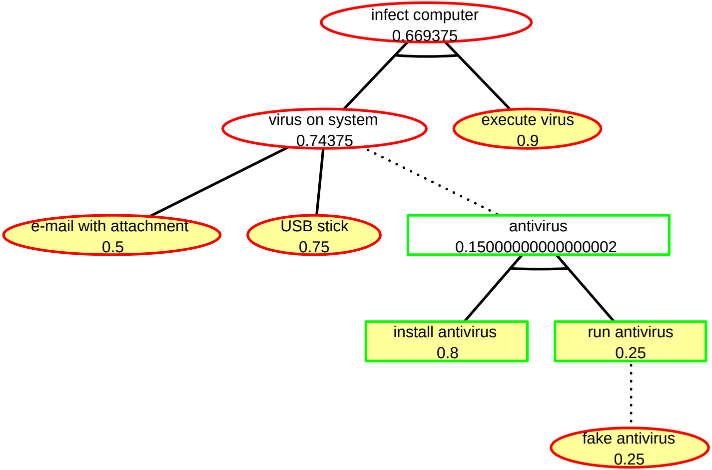
Probability of a countered subtree



$x(1 - y)$

Similarly for subtrees rooted in a defense node

Example: probability for infecting a computer



Limitations

The bottom-up procedure **does not take dependencies** between actions into account.

However, in practice

- Installing and running an antivirus
- Distributing and executing a virus

are **not independent actions**.

Thus, the standard bottom-up evaluation **is not suitable** for probabilistic assessment of attack–defense trees.

Challenges

1. How to design the **appropriate formalism**?
2. How to ensure that calculations **reflect the reality**?
3. How to guarantee the **efficiency** of the evaluation?

Proposed Framework [INS'16]

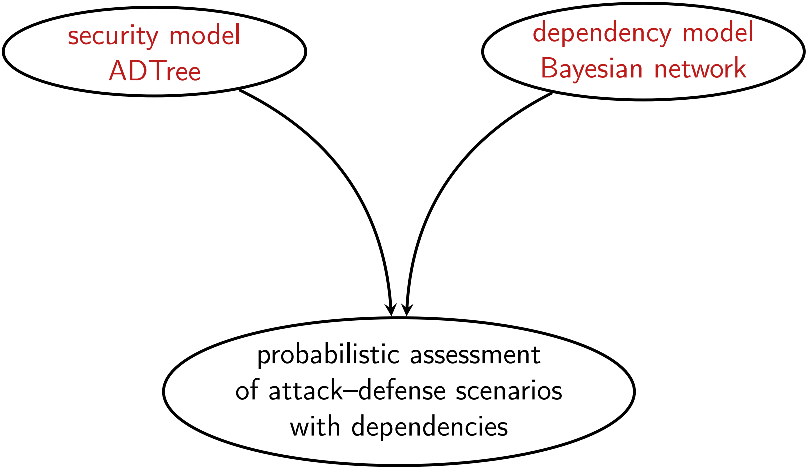
security model
ADTree

Proposed Framework [INS'16]

security model
ADTree

dependency model
Bayesian network

Proposed Framework [INS'16]



Modeling probability of dependent actions

Bayesian network

A directed, acyclic graph that reflects the conditional interdependencies between variables associated with the nodes of the network

Dependent variables

Conditional probability table for Y

$$p(Y = 1|X = 1) = 0.7$$
$$p(Y = 1|X = 0) = 0.2$$
$$p(Y = 0|X = 1) = 0.3$$
$$p(Y = 0|X = 0) = 0.8$$

Constructing Bayesian network BN_t for ADTree t

From an ADTree

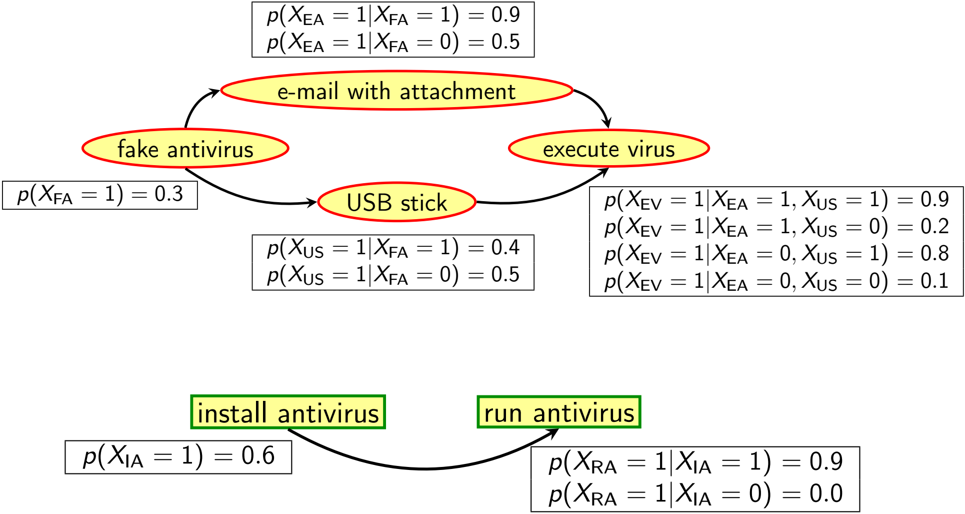
t – ADTree

$\mathcal{B}$ – set of all non-refined nodes of t

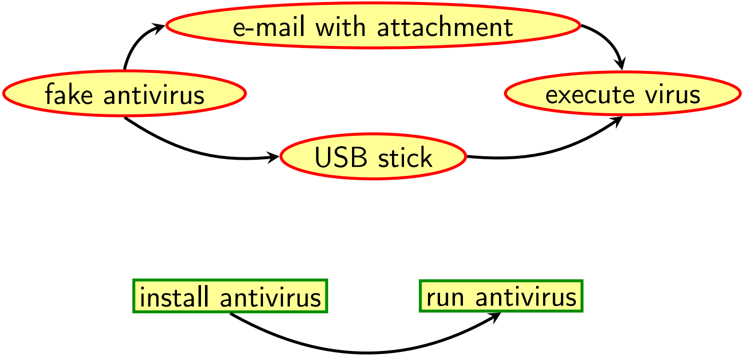
To a Bayesian network

- Elements of $\mathcal{B}$ are nodes of the Bayesian network BN_t
- Relations between actions are depicted by edges in BN_t
- Conditional probability tables quantify dependencies between actions

Example: BN_t for infecting a computer ADTree



Joint probability distribution for network BN_t



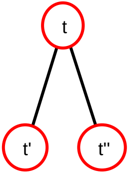
$$p(X_{\mathrm{EA}}, X_{\mathrm{US}}, X_{\mathrm{IA}}, X_{\mathrm{RA}}, X_{\mathrm{FA}}, X_{\mathrm{EV}}) =$$
$$p(X_{\mathrm{EV}}|X_{\mathrm{EA}}, X_{\mathrm{US}}) \times p(X_{\mathrm{EA}}|X_{\mathrm{FA}}) \times p(X_{\mathrm{US}}|X_{\mathrm{FA}}) \times p(X_{\mathrm{FA}}) \times p(X_{\mathrm{RA}}|X_{\mathrm{IA}}) \times p(X_{\mathrm{IA}})$$

Propositional semantics using algebraic operations

Non-refined

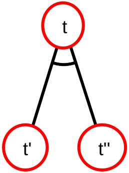
$f_t(A) = A$

OR



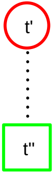
$f_t = f_{t'} \vee f_{t''}$

AND



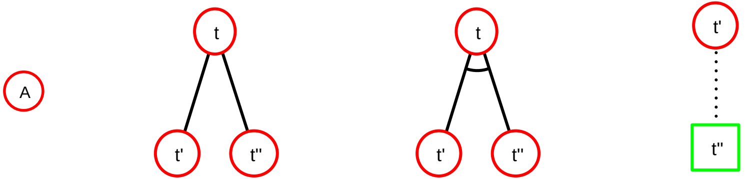
$f_t = f_{t'} \wedge f_{t''}$

Countermeasure



$f_t = f_{t'} \wedge \neg f_{t''}$

Propositional semantics using algebraic operations



Non-refined	OR	AND	Countermeasure
$f_t(A) = A$	$f_t = f_{t'} \vee f_{t''}$	$f_t = f_{t'} \wedge f_{t''}$	$f_t = f_{t'} \wedge \neg f_{t''}$
id_A	$\max\{f_{t'}, f_{t''}\}$	$f_{t'} \times f_{t''}$	$f_{t'} \times (1 - f_{t''})$

Probability computation

$\mathbf{x} \in \{0,1\}^{\mathcal{B}}$ – vector of successful/unsuccessful actions

Probability of attack vector $\mathbf{x}$

$$f_t(\mathbf{x}) \times p(\mathbf{x})$$

Probability related to ADTree t

$$P(t) = \sum_{\mathbf{x} \in \{0,1\}^{\mathcal{B}}} f_t(\mathbf{x}) \times p(\mathbf{x})$$

Probability of the most probable attack vector

$$P_{\max}(t) = \max_{\mathbf{x} \in \{0,1\}^{\mathcal{B}}} f_t(\mathbf{x}) \times p(\mathbf{x})$$

Compatibility results

Theorem

Probability computations on propositionally equivalent ADTrees yield the same result.

Observation

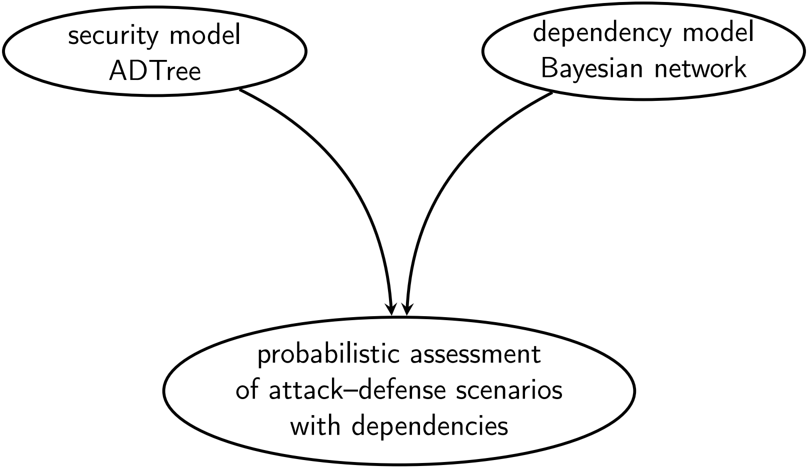
For ADTree t without dependent actions, $P(t)$ coincides with the result of the bottom-up computation.

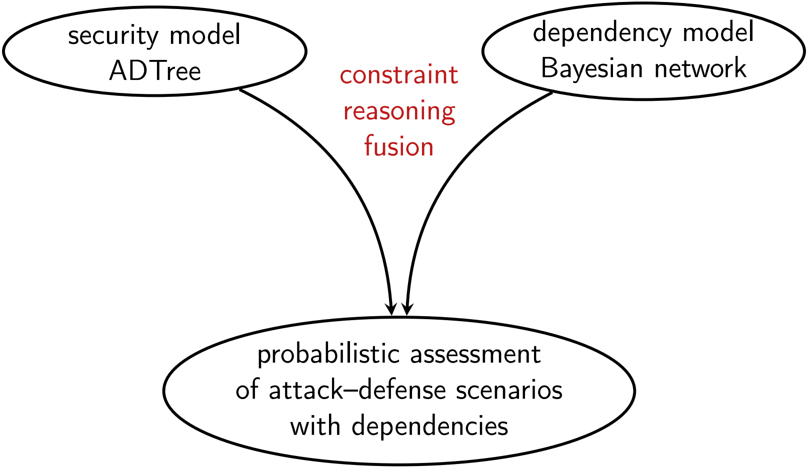
Efficiency problems

$$P(t) = \sum_{\mathbf{x} \in \{0,1\}^{\mathcal{B}}} f_t(\mathbf{x}) \times p(\mathbf{x}) \qquad P_{\max}(t) = \max_{\mathbf{x} \in \{0,1\}^{\mathcal{B}}} f_t(\mathbf{x}) \times p(\mathbf{x})$$

The **number of configurations x grows exponentially** with the number of involved actions. For large systems, it is therefore not feasible to

- Enumerate all the values of f_t
- Enumerate all the values of the joint probability distribution for BN_t


security model
ADTree
dependency model
Bayesian network
probabilistic assessment
of attack–defense scenarios
with dependencies


security model
ADTree
constraint
reasoning
fusion
dependency model
Bayesian network
probabilistic assessment
of attack–defense scenarios
with dependencies

Local indicators

$$f_t = \underbrace{\Big(\underbrace{(X_{\mathrm{EA}} \vee X_{\mathrm{US}})}_{Y_1} \wedge \neg \underbrace{\big(X_{\mathrm{IA}} \wedge \underbrace{(X_{\mathrm{RA}} \wedge \neg X_{\mathrm{FA}})}_{Y_2}\big)}_{Y_3}\Big)}_{Y_4} \wedge X_{\mathrm{EV}}$$

with the whole expression labelled Y_t:

$$\underbrace{\Big((X_{\mathrm{EA}} \vee X_{\mathrm{US}}) \wedge \neg \big(X_{\mathrm{IA}} \wedge (X_{\mathrm{RA}} \wedge \neg X_{\mathrm{FA}})\big)\Big) \wedge X_{\mathrm{EV}}}_{Y_t}$$

$$\begin{array}{lll}
\phi_1(Y_1, X_{\mathrm{EA}}, X_{\mathrm{US}}) = 1 & \text{exactly if} & Y_1 = \max\{X_{\mathrm{EA}}, X_{\mathrm{US}}\} \\
\phi_2(Y_2, X_{\mathrm{RA}}, X_{\mathrm{FA}}) = 1 & \text{exactly if} & Y_2 = X_{\mathrm{RA}} \times (1 - X_{\mathrm{FA}}) \\
\phi_3(Y_3, X_{\mathrm{IA}}, Y_2) = 1 & \text{exactly if} & Y_3 = X_{\mathrm{IA}} \times Y_2 \\
\phi_4(Y_4, Y_1, Y_3) = 1 & \text{exactly if} & Y_4 = Y_1 \times (1 - Y_3) \\
\phi_5(Y_t, Y_4, X_{\mathrm{EV}}) = 1 & \text{exactly if} & Y_t = Y_4 \times X_{\mathrm{EV}}
\end{array}$$

Global indicator function ϕ_t for ADTree t

Domain of ϕ_t:

- Non-refined nodes of t
- Inner variables of all local indicators

Global indicator function ϕ_t = product of all local indicators ϕ_i

$$\phi_t(\overbrace{Y_1, Y_2, Y_3, Y_4, Y_t}^{\mathcal{Y}=\text{inner variables}}, \overbrace{X_{EA}, X_{US}, X_{IA}, X_{RA}, X_{FA}, X_{EV}}^{\mathcal{B}=\text{non-refined nodes}}) =$$
$$\phi_1(Y_1, X_{EA}, X_{US}) \times \phi_2(Y_2, X_{RA}, X_{FA}) \times \phi_3(Y_3, X_{IA}, Y_2) \times$$
$$\phi_4(Y_4, Y_1, Y_3) \times \phi_5(Y_t, Y_4, X_{EV})$$

Φ_t indicates valid assignments with respect to f_t

Important property

Theorem

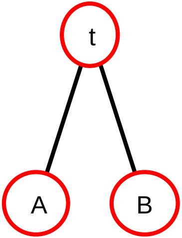
Consider an ADTree t over the set of non-refined nodes $\mathcal{B}$ and the global indicator function ϕ_t with the set of inner variables $\mathcal{Y}$.

$$\forall \mathbf{x} \in \{0,1\}^{\mathcal{B}} \;\; \exists! \mathbf{y} \in \{0,1\}^{\mathcal{Y}}, \quad \textit{such that} \;\; \phi_t(\mathbf{y}, \mathbf{x}) = 1$$

Corollary: $\forall \mathbf{x} \in \{0,1\}^{\mathcal{B}}$

$$\max_{\mathbf{y} \in \{0,1\}^{\mathcal{Y}}} \phi_t(\mathbf{y}, \mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^{\mathcal{Y}}} \phi_t(\mathbf{y}, \mathbf{x}) = 1$$

Filtering interesting assignments of ϕ_t



$$\phi_t(Y_t = 1, X_A = 1, X_B = 1) = 1$$
$$\phi_t(Y_t = 1, X_A = 1, X_B = 0) = 1$$
$$\phi_t(Y_t = 1, X_A = 0, X_B = 1) = 1$$
$$\phi_t(Y_t = 0, X_A = 0, X_B = 0) = 1$$

We are only interested in assignments such that $\phi_t = 1$ **and** $Y_t = 1$

$$Y_t \times \phi_t(\mathbf{y}, \mathbf{x})$$

Expressing f_t with its global indicator

$$\forall \mathbf{x} \in \{0,1\}^{\mathcal{B}}: \qquad \max_{\mathbf{y}\in\{0,1\}^{\mathcal{Y}}} \phi_t(\mathbf{y},\mathbf{x}) = \sum_{\mathbf{y}\in\{0,1\}^{\mathcal{Y}}} \phi_t(\mathbf{y},\mathbf{x}) = 1$$

$\forall \mathbf{x} \in \{0,1\}^{\mathcal{B}}$

$$\max_{\mathbf{y}\in\{0,1\}^{\mathcal{Y}}} \left(Y_t \times \phi_t(\mathbf{y},\mathbf{x})\right) = \sum_{\mathbf{y}\in\{0,1\}^{\mathcal{Y}}} \left(Y_t \times \phi_t(\mathbf{y},\mathbf{x})\right) =$$

$$= f_t(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ is an attack vector} \\ 0, & \text{otherwise} \end{cases}$$

Factorized form for probability formulas

Probability of attack vector $\mathbf{x}$

$$f_t(\mathbf{x}) \times p(\mathbf{x}) = \max_{\mathbf{y}\in\{0,1\}^{\mathcal{Y}}} \Big(Y_t \times \phi_t(\mathbf{y},\mathbf{x}) \times p(\mathbf{x}) \Big)$$

Probability related to ADTree t

$$P(t) = \sum_{\mathbf{x}\in\{0,1\}^{\mathcal{B}}} f_t(\mathbf{x}) \times p(\mathbf{x}) = \sum_{(\mathbf{y},\mathbf{x})\in\{0,1\}^{\mathcal{Y}\cup\mathcal{B}}} \Big(Y_t \times \phi_t(\mathbf{y},\mathbf{x}) \times p(\mathbf{x}) \Big)$$

Probability of the most probable attack vector

$$P_{\max}(t) = \max_{\mathbf{x}\in\{0,1\}^{\mathcal{B}}} f_t(\mathbf{x}) \times p(\mathbf{x}) = \max_{(\mathbf{y},\mathbf{x})\in\{0,1\}^{\mathcal{Y}\cup\mathcal{B}}} \Big(Y_t \times \phi_t(\mathbf{y},\mathbf{x}) \times p(\mathbf{x}) \Big)$$

Our framework in the context of semiring theory

- Inference problem over the arithmetic semiring $\langle \mathbb{R}, +, \times \rangle$

$$P(t) = \sum_{(\mathbf{y},\mathbf{x}) \in \{0,1\}^{\mathcal{Y} \cup \mathcal{B}}} \Big(Y_t \times \phi_t(\mathbf{y}, \mathbf{x}) \times p(\mathbf{x}) \Big)$$

- Inference problem over the product t-norm semiring $\langle [0,1], \max, \times \rangle$

$$P_{\max}(t) = \max_{(\mathbf{y},\mathbf{x}) \in \{0,1\}^{\mathcal{Y} \cup \mathcal{B}}} \Big(Y_t \times \phi_t(\mathbf{y}, \mathbf{x}) \times p(\mathbf{x}) \Big)$$

Local computation

Powerful local computation algorithms

- Fusion
- Variable elimination

$\Big\}$ **smart distributivity**

$P(t)$	Complexity bound	Using Nenok tool [IJAIT'10]
Direct computation	2^{11}	3.422sec
Using fusion	2^{4}	0.031sec

Complexity bounded by a **structural parameter** of the problem

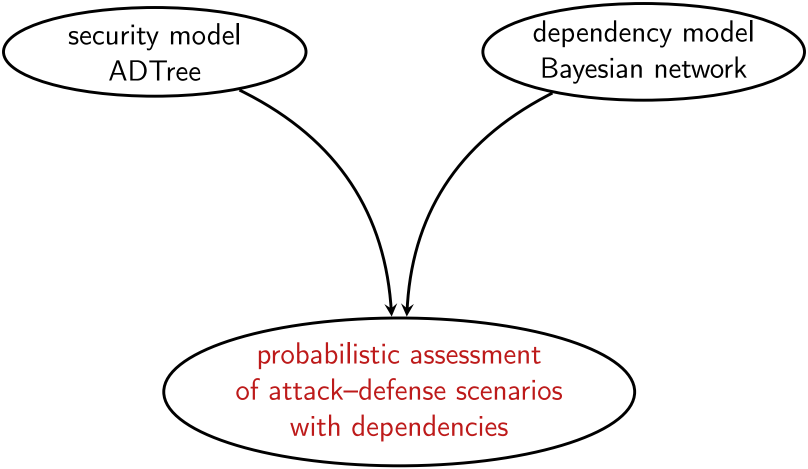
Summary

Summary

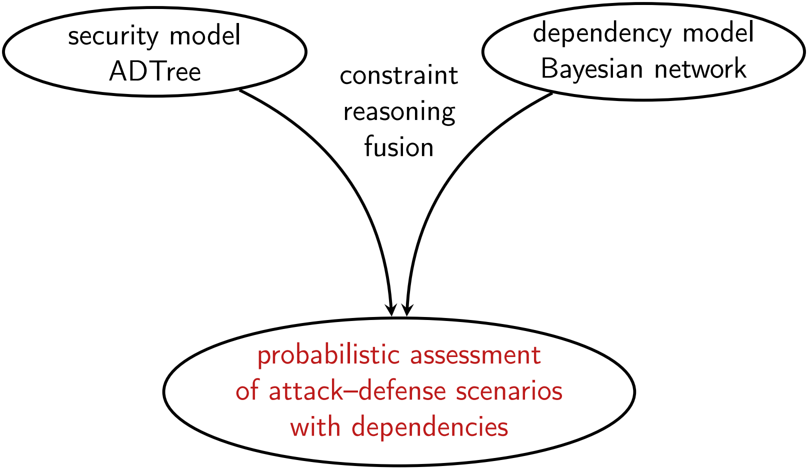
security model
ADTree

Summary

Summary


security model
ADTree
dependency model
Bayesian network
probabilistic assessment
of attack–defense scenarios
with dependencies

Summary


security model
ADTree
constraint
reasoning
fusion
dependency model
Bayesian network
probabilistic assessment
of attack–defense scenarios
with dependencies

Addressing challenges

1. How to design the **appropriate formalism**?
2. How to ensure that calculations **reflect the reality**?
3. How to guarantee the **efficiency** of the evaluation?

Addressing challenges

1. How to design the **appropriate formalism**?
   - Used by industry, intuitive & well formalized
   - Security model and dependency network are kept separated
2. How to ensure that calculations **reflect the reality**?
3. How to guarantee the **efficiency** of the evaluation?

Addressing challenges

1. How to design the **appropriate formalism**?
   - Used by industry, intuitive & well formalized
   - Security model and dependency network are kept separated
2. How to ensure that calculations **reflect the reality**?
   - Real-life data take dependencies into account
   - Complement ADTree with additional information
3. How to guarantee the **efficiency** of the evaluation?

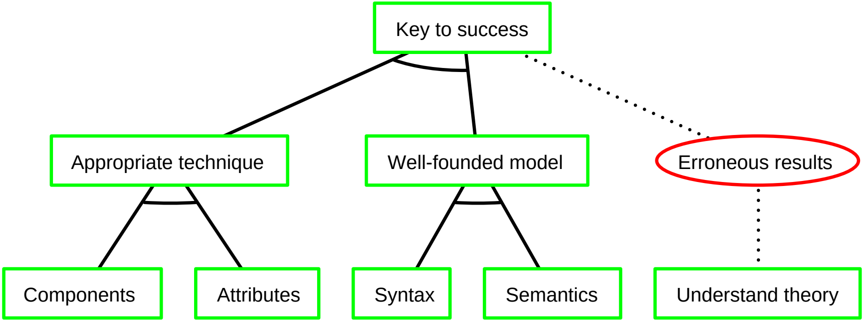
Addressing challenges

1. How to design the **appropriate formalism**?
   - Used by industry, intuitive & well formalized
   - Security model and dependency network are kept separated
2. How to ensure that calculations **reflect the reality**?
   - Real-life data take dependencies into account
   - Complement ADTree with additional information
3. How to guarantee the **efficiency** of the evaluation?
   - Local computation algorithms
   - Existing software tools, well-known heuristics

Where to take it from here?

- Find the best elimination sequence for Bayesian ADTrees
  - NP-complete in general
  - Prediction is possible for specific families of graphs
- Extend to probability distributions
  - Probability dependent on time
- Interface ADTool [QEST'13] with Nenok
  - Automated probability assessment of large scale scenarios

Take home message



References I

Barbara Kordy, Marc Pouly, and Patrick Schweitzer.
Probabilistic Reasoning with Graphical Security Models.
Information Sciences, Elsevier (to appear), 2016.

Barbara Kordy, Sjouke Mauw, Saša Radomirović, and Patrick Schweitzer.
Attack–Defense Trees.
Journal of Logic and Computation (JLC), 24(1):55–87, 2014.

Barbara Kordy, Ludovic Piètre-Cambacédès, and Patrick Schweitzer.
DAG-Based Attack and Defense Modeling: Don't Miss the Forest for the Attack Trees.
Computer Science Review, Elsevier, 13–14(0):1–38, 2014.

Marc Pouly.
Nenok - a software architecture for generic inference.
International Journal on Artificial Intelligence Tools, 19(1):65–99, 2010.

Barbara Kordy, Sjouke Mauw, and Patrick Schweitzer.
Quantitative Questions on Attack–Defense Trees.
In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *Information Security and Cryptology (ICISC 2012)*, volume 7839 of *LNCS*, pages 49–64. Springer, 2013.

Barbara Kordy, Marc Pouly, and Patrick Schweitzer.
A Probabilistic Framework for Security Scenarios with Dependent Actions.
In *Integrated Formal Methods (iFM 2014)*, LNCS, pages 256–271. Springer, 2014.

References II

Barbara Kordy, Marc Pouly, and Patrick Schweitzer.
Computational Aspects of Attack–Defense Trees.
In P. Bouvry, M. A. Klopotek, F. Leprevost, M. Marciniak, A. Mykowiecka, and H. Rybinski, editors, *Security & Intelligent Information Systems (SIIS 2011)*, volume 7053 of *LNCS*, pages 103–116. Springer, 2012.

Barbara Kordy, Piotr Kordy, Sjouke Mauw, and Patrick Schweitzer.
ADTool: Security Analysis with Attack–Defense Trees.
In Kaustubh R. Joshi, Markus Siegle, Mariëlle Stoelinga, and Pedro R. D'Argenio, editors, *Quantitative Evaluation of Systems (QEST 2013)*, volume 8054 of *LNCS*, pages 173–176. Springer, 2013.

Barbara Kordy, Sjouke Mauw, Matthijs Melissen, and Patrick Schweitzer.
Attack–Defense Trees and Two-Player Binary Zero-Sum Extensive Form Games Are Equivalent.
In Tansu Alpcan, Levente Buttyán, and John S. Baras, editors, *Decision and Game Theory for Security (GameSec 2010)*, volume 6442 of *LNCS*, pages 245–256. Springer, 2010.

Alessandra Bagnato, Barbara Kordy, Per Håkon Meland, and Patrick Schweitzer.
Attribute Decoration of Attack–Defense Trees.
International Journal of Secure Software Engineering (IJSSE), 3(2):1–35, 2012.

Bruce Schneier.
Attack Trees.
Dr. Dobb's Journal of Software Tools, 24(12):21–29, 1999.