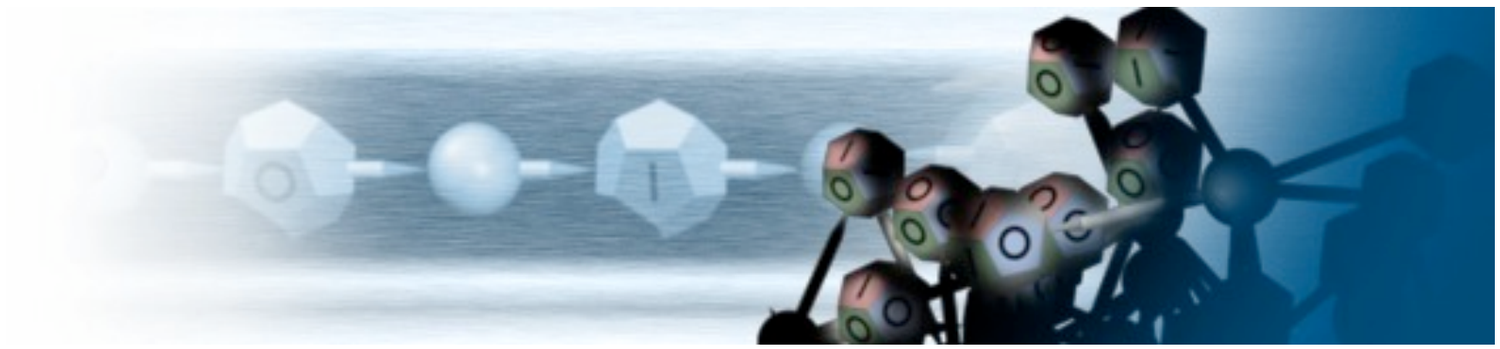
Syncrosoft
crypto & security solutions

Software Integrity based on the Problem of large Finite Automata Decomposition

Wulf Harder,
Re-Trust Workshop September 2009



Software Integrity

Syncrosoft
crypto & security solutions

- What are "Multi-Channel Finite Automata"? See the presentation at Re-Trust in March 2008: http://re-trust.dit.unitn.it/files/20080311Doc/harder-Syncrosoft-MCFACT.pdf

- How to compose Multi-Channel Finite Automata, see the animation: http://re-trust.dit.unitn.it/files/20090930Doc/presentations/MCFACTCompositionExample.avi

Software Integrity

Syncrosoft
crypto & security solutions

• Software integrity will be achieved by verifying data integrity of the input data for each instruction of the software. In case of integrity violation the data that are processed by the software are corrupted.

• The data processing instructions (not control-flow instructions) are transformed into atomic instructions. Atomic instructions are simple arithmetic instructions like addition/subtraction and logical operations. Multiplication/division are not atomic instructions and have to be replaced by sequences of atomic instructions.

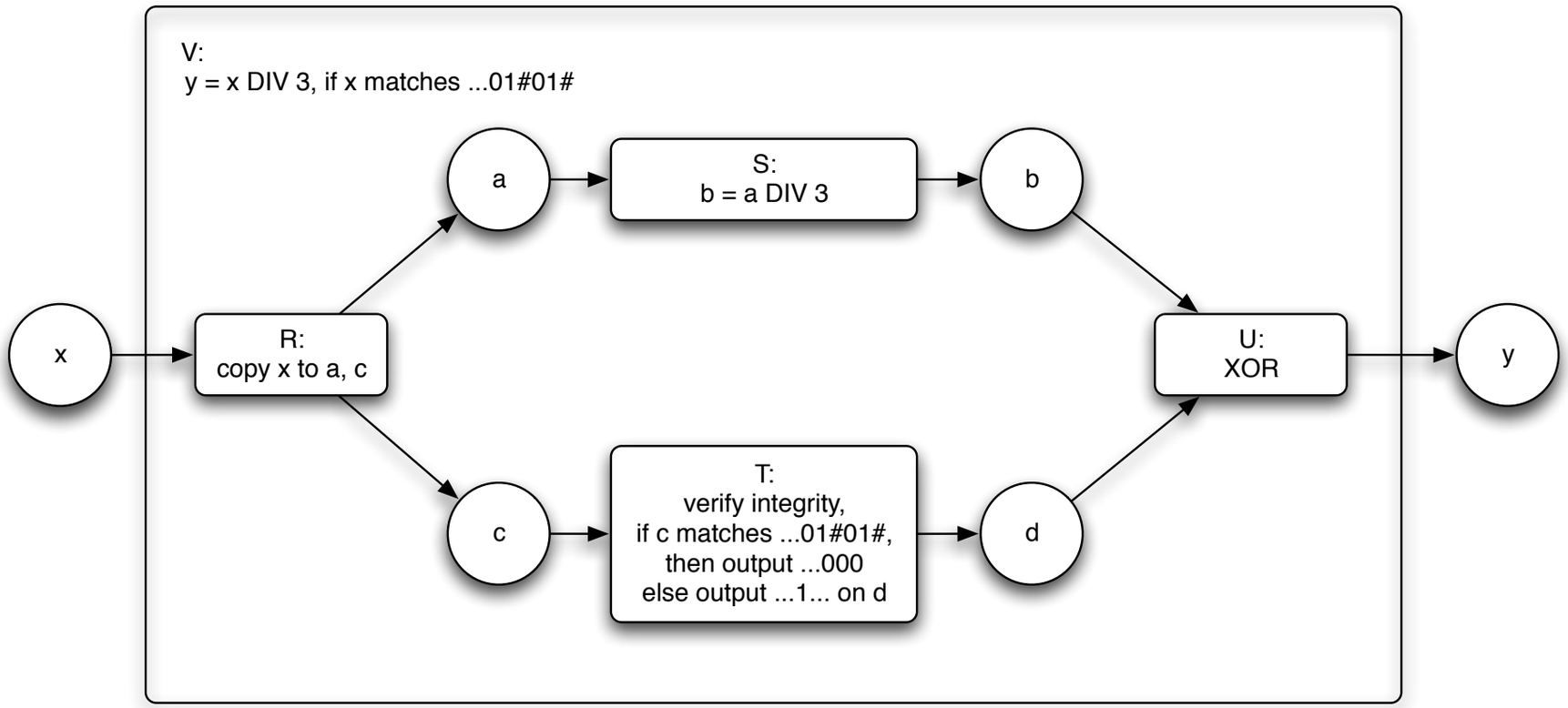
• All atomic instructions of the software are transformed into finite automata. The data that are processed by each instruction are verified for compliance with free definable rules. A rule could be e.g. that an instruction input has to be a word of a given regular language. This verification is performed by a finite automaton, which will be composed in parallel with the finite automaton of the instruction.

• Sequences of regular language verifications can emulate context-sensitive language verifications. Control-flow integrity can be assured by composing control-flow verifications in parallel with the finite automata of the instructions.

• The following slides demonstrate how to implement data integrity verification for an atomic instruction S. The finite automaton of the instruction S is embedded into a finite automaton V.

Software Integrity

The structure of V:

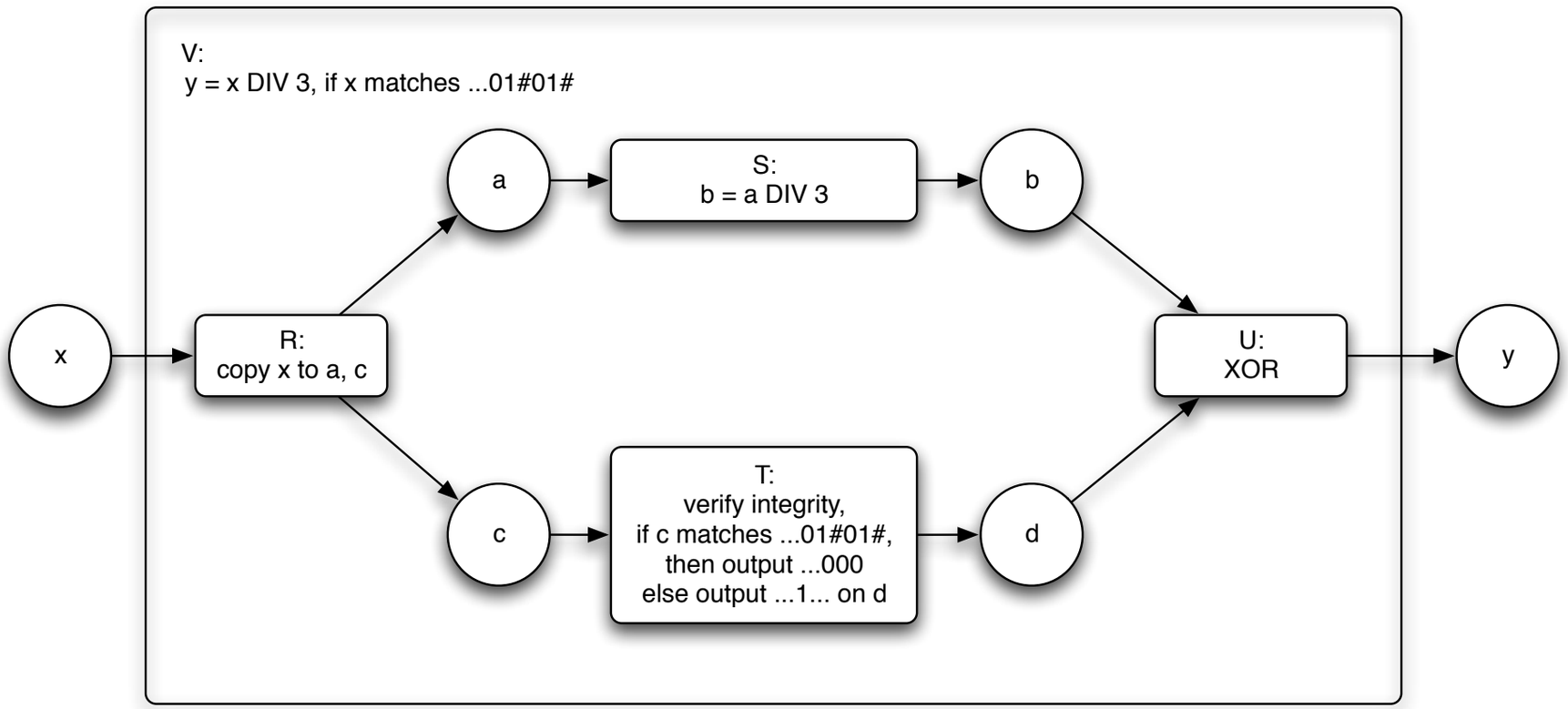


- This Petri-Net shows the structure of the composed finite automaton (transducer) V.
- R passes the input x to S and T.
- S divides its input by 3 (S can be used for multiplying numbers by 3 as well: commute input and output channel).

Software Integrity

Syncrosoft
crypto & security solutions

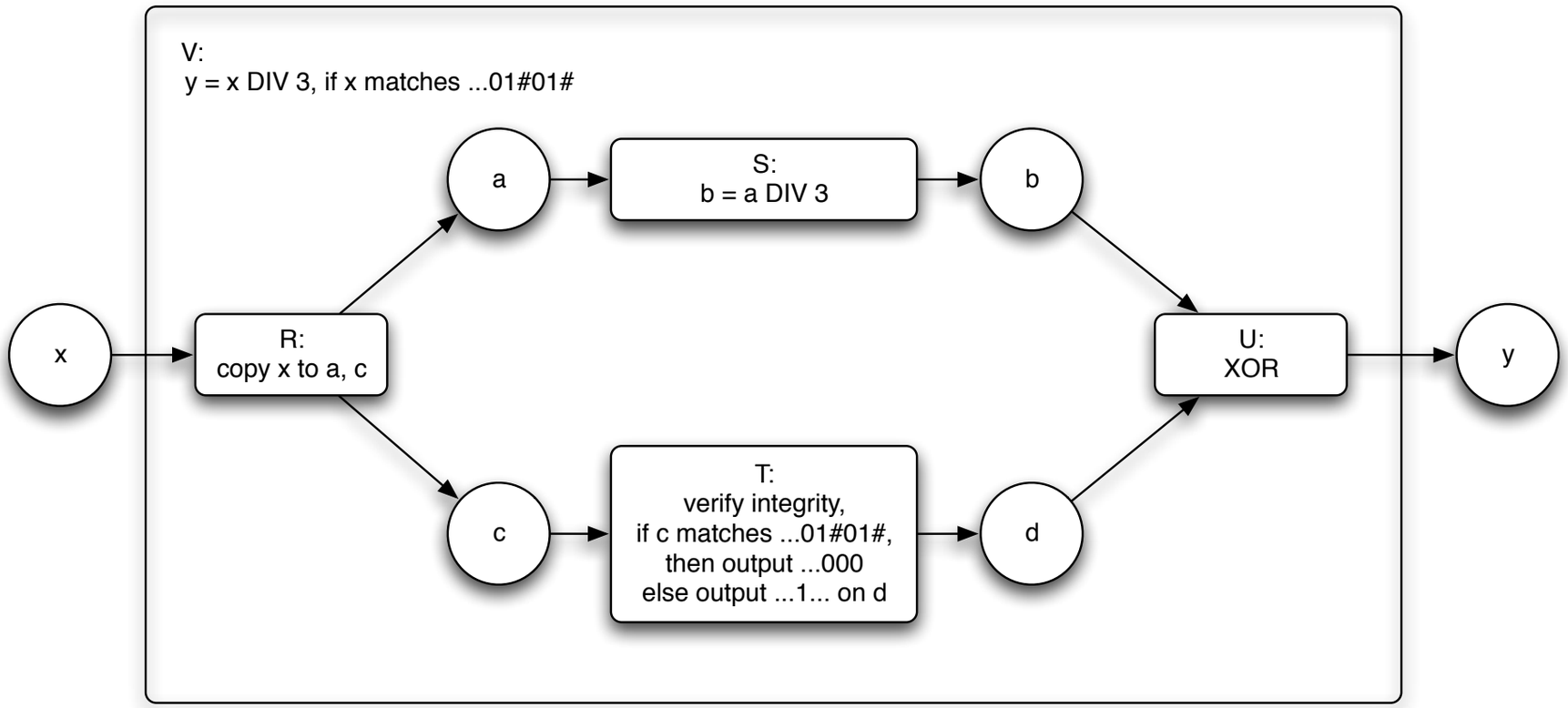
The structure of V:



- T verifies the integrity of its input.
- The integrity rule is arbitrary.

Software Integrity

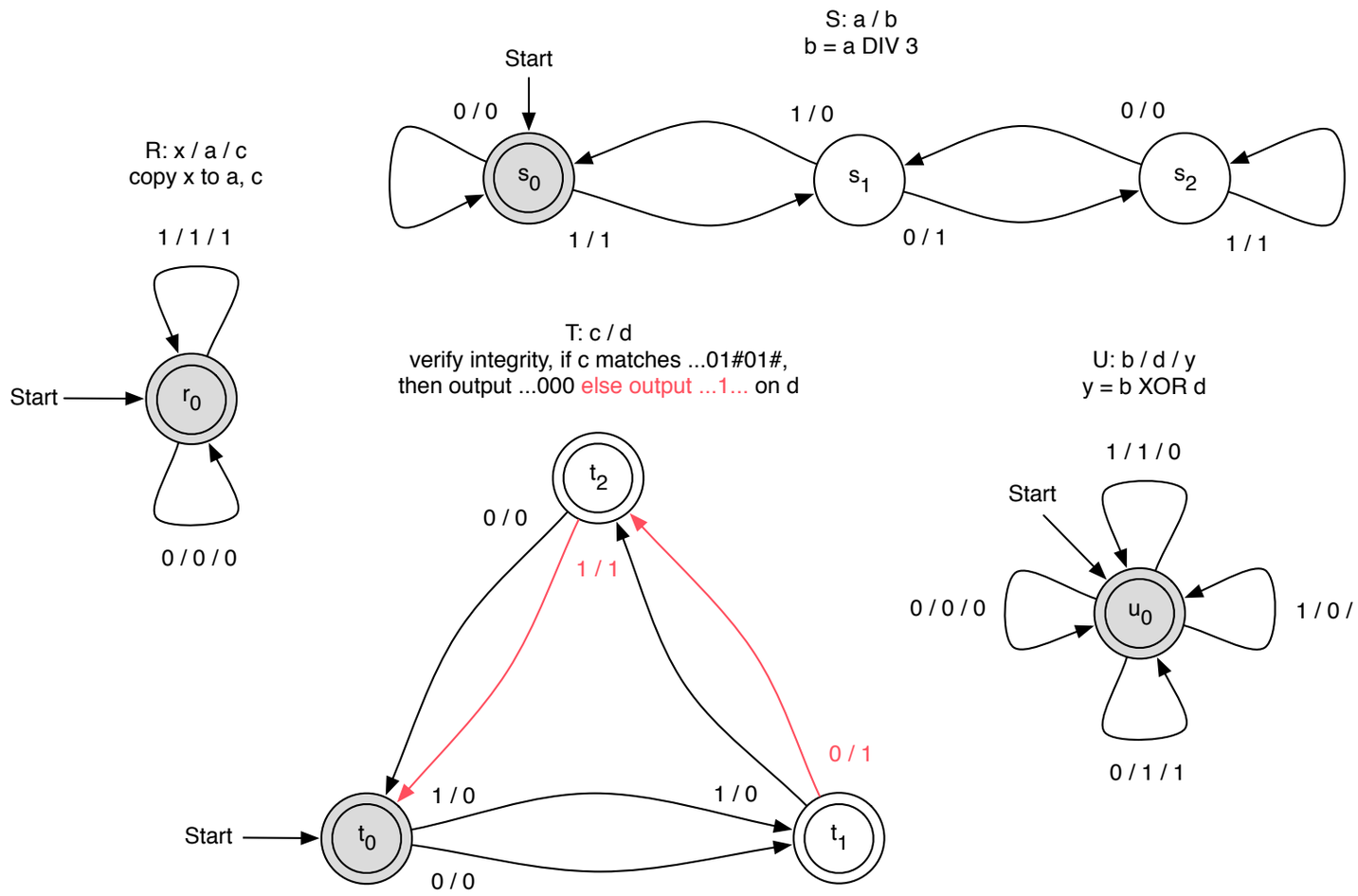
The structure of V:



- U XORs the outputs of S and T.

Software Integrity

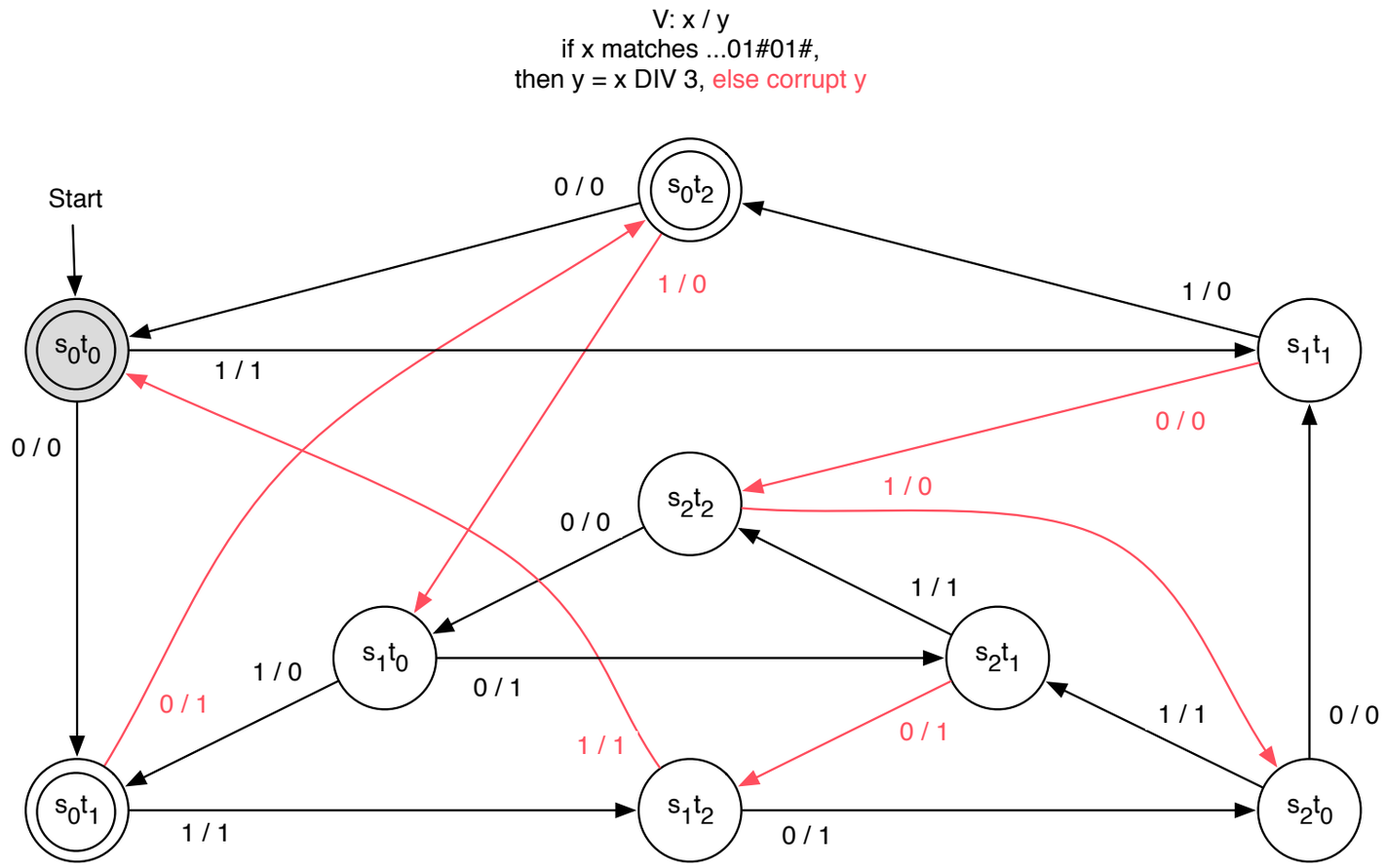
The components R, S, T, U:



- The input word must be a word of the language L = ((1+λ)(0+1)+λ)(01(0+1)).
- λ is the empty string and words have to be read from the right to the left (this is appropriate for words that represent numbers).
- Only black transitions of T produce words of L.

Software Integrity

The composition V:



- Decomposing large finite automata is a hard problem.
- A decomposition of V is prevented by composing V with a large finite automaton (transducer, e.g. a stream cipher), which encrypts the output of V.

Thank You!