

Exploiting Transient Behavior as Birthmarks of Combinational Circuits

Quentin TUAL

Lab-STICC

Université de Bretagne Sud

Lorient, France

quentin.tual@univ-ubs.fr

Jean-Christophe LE LANN

Lab-STICC

ENSTA

Brest, France

jean-christophe.le_lann@ensta.fr

Philippe COUSSY

Lab-STICC

Université de Bretagne Sud

Lorient, France

philippe.coussy@univ-ubs.fr

Abstract—The growing complexity of modern integrated circuits calls for robust characterization methods for verification and hardware security. Transient Behavior (TB) of synchronous designs is commonly used for power estimation and optimization, but may also support verification and security. Because TB reflects the circuit properties (structure, implemented logic, and technology-specific gate delays), it encodes the underlying physical and logical constraints. These constraints deterministically shape the timings of internal signals, allowing TB to act as a unique fingerprint for combinational circuits. Based on this idea, we introduce In-Cycle Transitions Distribution (ICTD), which captures the distribution of signal transition times within a representative clock cycle and provides a birthmark. Experiments on 1504 related and unrelated designs show that the resulting signatures are collision-free and highly sensitive to structural changes; in particular, inserting a single buffer modeled as a Hardware Trojan (HT) is detected in 100% of real application cases.

I. INTRODUCTION

Since the advent of semiconductor technology and the development of ICs, TB has been essentially exploited to estimate or optimize power consumption. TB arises from the capacitive and inductive properties of transistors and wires, which delay transitions and cause glitches before signals reach a stable state. In synchronous logic circuits, TB is deliberately avoided using clock signals in conjunction with flip-flops. The sequence of successive and non-stable values that occur between two sampling instances, most commonly between two sequential rising edges of the clock, constitutes what is referred to as TB.

Given that delays within a circuit arise from its components and their interconnections (transistors and wires), TB of a circuit is fundamentally a consequence of its combinational structure. Consequently, even minor modifications to the combinational structure, such as inserting a single gate, will unavoidably change its TB. In contrast, two entirely identical combinational structures should inherently exhibit similar TBs. Hence, we postulate that TB encapsulates the distinctive structural properties of the circuit and can serve as a unique signature for verification or security.

To verify the formulated hypothesis, we introduce a measure that can be used as a birthmark. The presented work reveals that TB can be leveraged as a signature of the structure of combinational circuits. This measure requires a dynamic analysis of the combinational circuit to obtain its birthmark.

Our contributions are as follows.

- We show that information about a combinational circuit exists in its own TB
- We show that TB can serve as a birthmark of the structure of a combinatorial circuit
- We propose a combinational circuit birthmarking method based on TB

This paper is organized as follows. Section II discusses context and related works. Section III defines the TB and the concepts used in the next sections, with pedagogical examples. Section IV gives precision on the proposition and explains the core idea. Section V presents the experimental evaluation. Section VI provides a summary and discusses future work.

II. BACKGROUND AND RELATED WORKS

In CMOS technologies, dynamic power stems from signal toggling, and glitches can account for a large share of it, motivating TB-based estimation [1][2][3] and glitch-reduction techniques such as gate freezing and RTL optimization [4][5][6]. However, transient values are rarely exploited directly, except in cyclic combinational circuits [7]. Sensitizing signals has also been studied to generate fault-uncovering test patterns [8][9].

HT detection is classically addressed by side-channel and delay fingerprinting [10][11]. Few works have implicitly leveraged TB [12][13] by operating the circuit in a dysfunctional mode and using TB values as features in an ML model: [12] overclocks the sequential circuit, while [13] reduces voltages, magnifying the effect of a potentially inserted HT on the TB. Circuit watermarking [14][15] and Physical Unclonable Function (PUF) exploiting TB like [16][17] are other directions related to birthmarking against IP piracy.

In this paper, a new measure is proposed that exploits the TB of a combinational circuit operating under normal conditions. It signs the circuit structure using a measure similar to Switching Activity (SA) but time-aware, providing significant additional insight. Unlike the PUF proposed in [16], no logic insertion is required to sign the circuit, only its own structure is used. Unlike overclocking methods [12][13], the circuit remains in normal operation mode at all steps of the proposed birthmarking method. The TB and the associated concepts underlying the proposed method are presented in the next section.

III. TRANSIENT BEHAVIOR

The TB is often analyzed macroscopically to estimate power consumption, for example using Switching Activity (SA). The first subsection defines SA and describes how to compute it probabilistically via static analysis. With non-zero delay models, additional transitions appear on internal signals and outputs after input updates, creating TB. The second subsection explains how these transitions propagate through the circuit.

A. Switching activity

The SA of a combinational circuit refers to the average number of transitions per clock cycle at a gate or primary output. Assuming a zero delay model, the SA of a signal S can be probabilistically defined as $SA(S) = P(S_t \neq S_{t+1})$, with S_t the value of S at the instant t . Let $y = (a + b) \cdot c$ be a combinational circuit. It is assumed that $P(a = 1) = p_a$, $P(b = 1) = p_b$ and $P(c = 1) = p_c$ are the respective and independent probabilities that the inputs a , b and c are equal to 1. The probability $P(y = 1)$ that the output y is equal to 1 and the SA of y computed probabilistically are given below.

$$\begin{aligned} P(a + b = 1) &= 1 - (1 - p_a)(1 - p_b) \\ P(y = 1) &= P(a + b = 1) \cdot p_c \\ SA(y) &= 2 \cdot P(y = 1) \cdot (1 - P(y = 1)) \end{aligned}$$

For example, with $p_a = p_b = p_c = 0.5$, $SA(y) = 0.46875$.

When it is considered that logic gates have a non-zero delay model, TB emerges, which the probabilistic approach does not account for. Including TB in the probabilistic evaluation of SA would require estimating the transition probability at every instant of the TB, which would greatly increase the complexity of the previous equations. Consequently, dynamic analysis (i.e. simulation) is used to obtain TB and compute SA.

B. Delay model and transition propagation

In a non-zero inertial delay model, changes at the inputs do not always result in transitions at the outputs. Output transients depend on the circuit structure, each gate's logic and delay, and the specific input transitions. More specifically, a transition may not propagate for the two reasons listed below.

- *Logical masking*: A controlling value on one gate input blocks transitions on the other inputs from propagating.
- *Inertial masking*: The time between transitions on the gate inputs is shorter than the gate inertial delay.

In the first case (logical masking), the probability of a controlling value on one input increases with the number of inputs. It is shown with the following formalization, where g is a gate, E is the event of at least one input of g showing a controlling value, $\bar{E}$ is the opposite event where no inputs of g show a controlling value. A controlling value on an input x is represented as $ctl(x)$ and $I(g)$ is the set containing all the inputs of g . Using this formalization, we can say:

$$P(E) = 1 - P(\bar{E}) = 1 - \prod_{\forall x \in I(g)} 1 - P(ctl(x))$$

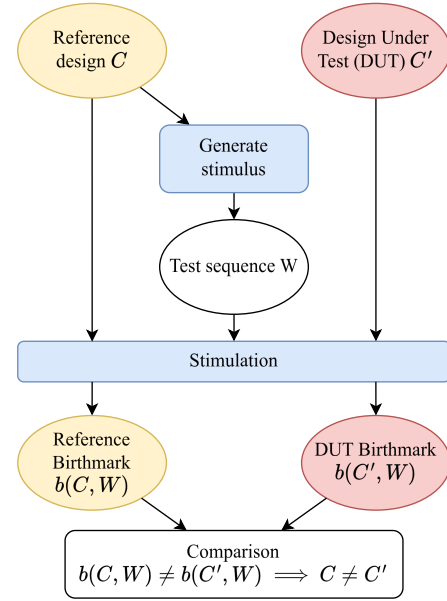


Fig. 1: Flow of the method

If we consider $P(ctl(x)) = 0.5$, $\forall x \in I(g)$, then we know that $P(E) = 0.75$ for a two input gate and $P(E) \approx 0.96$ for a five input gate. Thus, it confirms that $P(E)$ increases with $|I(g)|$.

The second cause (inertial masking) depends on the spread of delays in the model. If all delays are equal (uniform delay model), inertial masking never occurs. As the gate-delay variation increases, the probability of inertial masking increases. This variation in gate delays can be quantified as the average distance between gate delays μ , for a given delay model and gate library. For a gate g within the gate library G (a set of gate types), its delay is denoted $\delta(g)$. A set containing all possible pairs of two different gates from G is denoted H such that $H = \{(g_a, g_b), \forall g_a, g_b \in G, \text{ with } g_a \neq g_b\}$. Therefore, $\mu = \frac{s}{|H|}$ with s the sum of the distances of each pair in H , written $s = \sum_{\forall (g_a, g_b) \in H} |g_a - g_b|$. For six gates with identical delays, $\mu = 0$. However, if three gates have delay 1 and the other three delay 2, then $\mu = 0.6$. For a given gate library, we thus quantify the average distance between gate delays, indicating the likelihood of inertial masking.

Both described masking effects also appear with more realistic delay models and in real circuits. These masking phenomena are important in the analysis of TB for structure signatures because they block transitions and reduce observable activity. They cause limitations in the proposed birthmarking technique, as discussed in Section V. The birthmarking method itself is detailed in the next section.

IV. METHODOLOGY

In this section, we introduce a method that signs the structure of a combinational circuit C with its TB, producing a birthmark $b(C, W)$, where W is a test sequence. C includes all gate interconnections, size, and delay details. This birthmark enables both stolen Intellectual Property (IP) identification and HT detection by comparison with a golden reference (see Fig. 1).

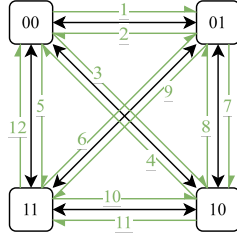


Fig. 2: A complete directed graph G built from the set V .

The first subsection defines the exhaustive transition test pattern that activates all input transitions of the target circuit; this pattern is later used in Section V to evaluate the birthmark. The second subsection explains dynamic SA computation, also used in Section V. The final subsection presents the proposed ICTD birthmarking approach, derived from TB analysis.

A. Exhaustive stimulation in transition

To test a combinational circuit corresponding to the combinational layer of a sequential circuit, synchronous stimulation is used: all inputs are updated simultaneously on the rising edge of the clock. Exhaustive transition-based stimulation of the circuit explores all possible TBs under synchronous operation, unlike exhaustive value-based stimulation, which only explores all possible stable output values. Let a combinational circuit have N inputs. The exhaustive set of input values is $V = \{0, \dots, 2^N - 1\}$. An exhaustive transition sequence W consists of all possible ordered pairs of values from V . To construct such a sequence, one can use graph theory principles. Let G be a complete directed graph where each node represents a value in V . The goal is to find an Eulerian circuit in G , i.e., a path that traverses each directed edge exactly once and returns to the starting node. For example, with $N = 2$, V and W are:

$$V = \{0, 1, 2, 3\} \quad W = (0, 1, 0, 2, 0, 3, 1, 2, 1, 3, 2, 3, 0)$$

The corresponding graph is shown in Fig. 2 where a Eulerian circuit is shown with numbered arrows. This generation method allows us to build a stimulation sequence that is exhaustive in transitions. However, the transition-exhaustive test does not scale well, since the sequence W contains $2^{2N} - 2^N + 1$ test vectors, where N is the number of circuit inputs. Therefore, it is mainly useful in experiments to fully characterize the TB of circuits of limited size.

B. Switching Activity measurement

SA is the number of transitions per cycle on a given signal. Section III shows how it can be derived probabilistically under a zero-delay model. In this work, it is calculated dynamically via circuit simulation, so that delays from a non-zero delay model are considered. For an input sequence W and a logic cone output o , the following steps are applied.

- Simulate the circuit over $|W|$ cycles applying W .
- Count the number of transitions N_{trans} observed on o .
- Compute the switching activity (SA) as follows.

$$SA(o) = \frac{N_{\text{trans}}}{|W|}$$

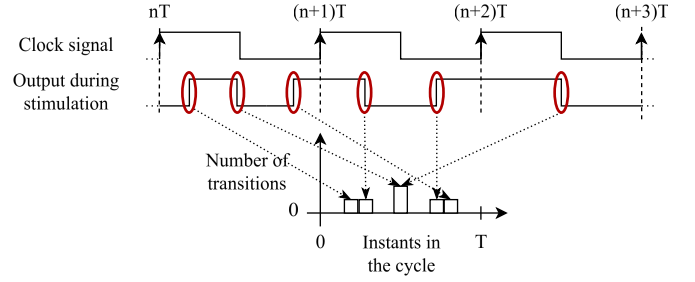


Fig. 3: Measure of the distribution of the transitions inside the cycle

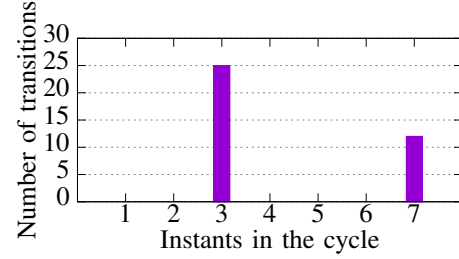


Fig. 4: ICTD for the example circuit of the Section III.A, obtained using exhaustive test detailed in Section IV.A

Considering each gate delay being 1 in Arbitrary Unit (AU), $SA(y) \approx 0.649$ for the example discussed in Section III.

C. ICTD birthmark

The proposed birthmark extends beyond SA by adding temporal information. This measure records the transition counts at each instant in the cycle as visualized in Fig. 3. It is formally defined as $D(t) = n$, $\forall t \in [0, T]$ where T is the duration of the clock cycle, t an instant in the cycle, and n a number of transitions. The numbers of transitions observed at each instant in all cycles during the simulation are overlaid on a typical clock cycle. Circuits consist of several, more or less overlapped, logical cones. Each cone requires a distinct distribution measurement. Consequently, the birthmark $b(C, W)$ is a series of distributions, each associated with an output of C . Furthermore, in scenarios with realistic delay models, further parameters could impact $b(C, W)$, accounting for variations in gate delays due to manufacturing or environment during operation. The ICTD obtained for the single output of the example circuit from Section III is shown in Fig. 4.

In the next section, exhaustive transition stimulation is used to evaluate the proposed ICTD measure as a birthmark on several different combinational circuits.

V. EXPERIMENTAL EVALUATION

The experiments evaluate how well the proposed birthmark uniquely characterizes a combinational structure. The first subsection outlines the experimental setup. The second subsection reports the results, organized into three experiments. The first compares signatures of circuits with substantially different structures. The second repeats this comparison for circuits with highly similar structures. The third examines the method's limitations to provide further insight into its behavior.

TABLE I: Circuits characteristics

Set	# circuits	# inputs	# outputs	# gates_CP
Random	288	8-12	4-8	6-10
LGSynth91	88	3-12	1-63	1-40

TABLE II: Gate delays

Gate type	NOT	AND	OR	NAND	NOR	XOR
β	1	1	1	2	2	3

A. Experimental setup

The experiments are performed in Ruby : scripting, generating VHDL descriptions of the combinational circuits and stimulation sequences. Simulations are performed through GHDL [18].

1) *Benchmark*: The experiments cover two sets of combinational circuits. The first group comprises 288 randomly generated circuits, whose main characteristics are summarized in the first row of Table I. The second group is selected from the LGSynth91 benchmark [19], with its characteristics reported in the second row of Table I. The LGSynth91 circuits are mapped to four distinct gate libraries with the ABC tool [20]. Each library G contains standard Boolean gates (NAND, NOR, NOT, OR, XOR, ...) whose input arity can range from 1 up to $A(G)$.

2) *Delay model*: The experiments use an integer inertial delay model. Each gate type is assigned a base delay $\beta(g)$ (reported in Table II), and the total delay $\delta(g)$ increases with the number of inputs according to $\delta(g) = \beta(g) + ||I(g)||$. For example, an AND2 gate has an inertial propagation delay of 3 AU, while an AND5 gate has a delay of 6 AU.

3) *Stimulation*: The circuits are stimulated with an exhaustive transition test according to the method described earlier in Section IV. To avoid an extremely long stimulation time, the circuits in both benchmarks are limited to a maximum of 12 inputs, which corresponds to almost 16 million test vectors.

B. Experiment 1: coincidence risk of ICTD birthmark

The ICTD of each circuit of both sets is compared with all others to evaluate the risks of coincidence (or collision), i.e., an identical signature for two different structures. The 288 randomly generated circuits and 88 real application circuits are mapped with each of the four gate libraries, in total, the birthmarks of each of 1504 circuits are compared by pair.

No coincidence has been encountered, indicating that each circuit has a unique signature. Therefore, the ICTD birthmark can serve as a unique signature for combinational circuits. The experimental results show that ICTD effectively distinguishes both vastly different structures (e.g., a synthetic circuit vs. a real circuit) and moderately different structures (e.g., the same circuit mapped with two different gate libraries).

C. Experiment 2: ICTD to detect small structural alterations

This experiment verifies whether the proposed birthmark can distinguish similar related structures by detecting minimal structural changes, such as HT. For each circuit in both benchmarks, we create an altered version by randomly inserting a buffer associated with an OR-gate delay. We measure the

TABLE III: Results of the Experiment 2 & 3

$A(G)$	μ	Random (288 circ.)		LGSynth91 (88 circ.)	
		Detection %	Avg. SA	Detection %	Avg. SA
2	1.62	95.83 %	2.85	100.00 %	1.99
3	1.60	89.58 %	2.66	100.00 %	1.80
4	1.75	82.64 %	2.68	100.00 %	1.72
5	1.97	73.61 %	2.47	100.00 %	1.71

ICTD for each initial and altered circuit version; any difference reveals buffer insertion. The delay model ensures deterministic TB in the circuits, excluding false positives.

The "Detection %" columns of Table III confirm that ICTD can uniquely sign the combinational structure and detect minimal structural alterations, achieving up to 95.83 % detection for the Random benchmark and 100 % for the LGSynth91 benchmark.

D. Experiment 3: Masking impact on ICTD

This experiment evaluates the impact of logical and inertial masking on TB. The SA and μ , as defined in Section III.B, are measured for several gate libraries, each characterized by its maximum gate arity $A(G)$. Increasing $A(G)$ increases μ , the average distance between gate delays. As discussed in Section III, higher $A(G)$ and μ increase the probability of logical and inertial masking, leading to less transient activity.

The masking effects are observable in the "Avg. SA" columns of Table III, where the average SA decreases as $A(G)$ increases for both random and real application benchmarks. A greater μ hinders transition propagation, reducing transient activity at the outputs and the information available to sign the circuit structure. Consequently, alteration detection becomes less effective as μ increases, with the detection rate falling from 95.83 % to 73.61 % for random circuits. For the LGSynth91 real-application circuits, all alterations are detected even at lower SA than in random circuits. This suggests that limitations come not only from the number of transitions in TB, but also from circuit structure. Structural optimizations and path balancing by the ABC tool during mapping may improve signal observability and controllability, increasing detection rates.

VI. CONCLUSION & DISCUSSION

This paper proposes ICTD, a combinational circuit birthmarking method based on TB. Experiments on 1504 circuits, from highly dissimilar to highly similar, show no birthmark collisions and that even minimal structural changes are always detected in real application circuits. These results confirm that TB carries enough structural information to uniquely sign combinational circuits using ICTD, which is valuable for hardware security (e.g. IP piracy and HT detection).

Future work will follow three main directions. First, we will address masking limitations by studying how Boolean optimization affects coincidence risks. Second, to improve scalability, we will design intelligent stimulus generation targeting relevant TBs. Third, we will adapt the method to more realistic delay models, including process variations.

REFERENCES

- [1] F. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, no. 4, pp. 446–455, Dec. 1994.
- [2] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," in *[1992] Proceedings 29th ACM/IEEE Design Automation Conference*. Anaheim, CA, USA: IEEE Comput. Soc. Press, 1992, pp. 253–259.
- [3] M. Meixner and T. G. Noll, "Accurate Estimation of CMOS Power Consumption Considering Glitches by Using Waveform Lookup," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 7, pp. 787–791, Jul. 2017.
- [4] A. Raghunathan, S. Dey, and N. K. Jha, "Glitch Analysis and Reduction in Register Transfer Level Power Optimization," *Proceedings of the 33rd annual Design Automation Conference*, pp. 331–336, 1996.
- [5] L. Benini, G. De Micheli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Glitch power minimization by selective gate freezing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 287–298, Jun. 2000.
- [6] S. Bathla, R. M. Rao, and N. Chandrathoodan, "A Simulation-Based Metric to Guide Glitch Power Reduction in Digital Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 2, pp. 376–386, Feb. 2019.
- [7] S. Malik, "Analysis of cyclic combinational circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 7, pp. 950–956, Jul. 1994.
- [8] H. Chang and J. A. Abraham, "VIPER: An efficient vigorously sensitizable path extractor," in *Proceedings of the 30th International on Design Automation Conference - DAC '93*. Dallas, Texas, United States: ACM Press, 1993, pp. 112–117.
- [9] M. Sauer, B. Becker, and I. Polian, "PHAETON: A SAT-Based Framework for Timing-Aware Path Sensitization," *IEEE Transactions on Computers*, vol. 65, no. 6, pp. 1869–1881, Jun. 2016.
- [10] J. Cruz and J. Hamlet, "A Survey on the Design, Detection, and Prevention of Pre-Silicon Hardware Trojans," *IEEE Access*, vol. 13, pp. 58 509–58 547, 2025.
- [11] F. N. Esirci and A. A. Bayrakci, "Delay based hardware Trojan detection exploiting spatial correlations to suppress variations," *Integration*, vol. 91, pp. 107–118, Jul. 2023.
- [12] X. Meng, R. Hassan, S. M. P. Dinakrrao, and K. Basu, "Can Overclocking Detect Hardware Trojans?" in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2021, pp. 1–5.
- [13] V. R. Surabhi, P. Krishnamurthy, H. Amrouch, J. Henkel, R. Karri, and F. Khorrani, "Exposing Hardware Trojans in Embedded Platforms via Short-Term Aging," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3519–3530, Nov. 2020.
- [14] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Watermarking Techniques for Intellectual Property Protection," *ACM/IEEE Design Automation Conference (DAC)*, 1998.
- [15] H. Badier, C. Pilato, J.-C. L. Lann, P. Coussy, and G. Gogniat, "Opportunistic IP Birthmarking using Side Effects of Code Transformations on High-Level Synthesis," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Grenoble, France: IEEE, Feb. 2021, pp. 52–55.
- [16] D. Suzuki and K. Shimizu, "The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes," in *Cryptographic Hardware and Embedded Systems, CHES 2010*, S. Mangard and F.-X. Standaert, Eds. Berlin, Heidelberg: Springer, 2010, pp. 366–382.
- [17] R. Yasaei, S.-Y. Yu, E. K. Naeini, and M. A. A. Faruque, "GNN4IP: Graph Neural Network for Hardware Intellectual Property Piracy Detection," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. San Francisco, CA, USA: IEEE, Dec. 2021, pp. 217–222.
- [18] T. Gingold, "GHDL, free and open source VHDL simulator," ghdl, Apr. 2025.
- [19] S. YANG, "Logic synthesis and optimization benchmarks user guide," *Technical Report 1991-IWLS-UG-Saeyang*, 1991.
- [20] R. Brayton and A. Mishchenko, "ABC: An Academic Industrial-Strength Verification Tool," in *Computer Aided Verification*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, T. Touili, B. Cook, and P. Jackson,

Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 6174, pp. 24–40.