Generalized Probabilistic Satisfiability and Applications to Modelling Attackers with Side-Channel Capabilities[☆]

Carlos Caleiro

SQIG - Instituto de Telecomunicações
Dep. Mathematics, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Filipe Casal

Centro de Matemática, Aplicações Fundamentais e Investigação Operacional (CMAF-CIO)
Dep. Mathematics, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Andreia Mordido

LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal

Abstract

We analyze a generalized probabilistic satisfiability problem (GenPSAT) which consists in deciding the satisfiability of linear inequalities involving probabilities of classical propositional formulas. GenPSAT is proved to be NP-complete and we present a polynomial reduction to Mixed-Integer Programming. Capitalizing on this translation, we implement and test a solver for the GenPSAT problem. As previously observed for many other NP-complete problems, we are able to detect a phase transition behaviour for GenPSAT. We also describe GGenPSAT, which generalizes GenPSAT by allowing Boolean combinations of linear inequalities involving probabilities of classical propositional formulas which we use to develop applications in information security. Namely, in the context of cryptographic protocols, we model classes of attackers with side-channel capabilities, and study the problem of deciding whether a formula is perfectly masked in the presence of such attackers.

Keywords: Probabilistic Satisfiability, GenPSAT, GGenPSAT, Phase Transition, Side-Channel Attacks

[☆]This paper extends the research presented in [1]. Work done under the scope of R&D Unit 50008, financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER–PT2020), partially supported by Fundação para a Ciência e a Tecnologia by way of grant UID/MAT/04561/2013 to Centro de Matemática, Aplicações Fundamentais e Investigação Operacional of Universidade de Lisboa (CMAF-CIO), and also supported by the LASIGE Research Unit, ref. UID/CEC/00408/2013. FC acknowledges the support from the DP-PMI and FCT (Portugal) through scholarship SRFH/BD/52243/2013.

1. Introduction

For many years, the satisfiability problem for propositional logic (SAT) has been extensively studied both for theoretical purposes, such as computational complexity theory, and for practical purposes. In spite of its NP-completeness [2], modern tools for solving SAT are able to cope with very large problems in a very efficient manner, leading to applications in many different areas and industries [3].

Naturally, people started extending this problem to more expressive frameworks: for instance in Satisfiability Modulo Theories [4], instead of working in propositional logic, one can try to decide if a formula is valid in some specific first-order theory. One other direction is to extend propositional logic with probabilities. The probabilistic satisfiability problem (PSAT) was originally formulated by George Boole [5] and later by Nilsson [6]. This problem consists in deciding the satisfiability of a set of assignments of probabilities to propositional formulas. There has been a great effort on the analysis of the probabilistic satisfiability problem and on the development of efficient tools for the automated treatment of this problem [7, 8, 9, 10, 11].

In this paper we study a Generalized Probabilistic Satisfiability problem (GenPSAT) extending the scope of PSAT by allowing linear combinations of probabilistic assignments of values to propositional formulas, with applications in the analysis of the security of cryptographic protocols and on estimating the probability of existence of attacks [12]. Intuitively, GenPSAT consists in deciding the existence of a probability distribution satisfying a set of classical propositional formulas with probability 1, and a set of linear inequalities involving probabilities of propositional formulas. The GenPSAT problem was previously identified in the context of the satisfiability of the probabilistic logic in [13], where it was also shown to be NP-complete. Here, we explore the computational behaviour of this problem and present a polynomial reduction from GenPSAT to Mixed-Integer Programming, following the lines of [9, 10].

Mixed-Integer Programming (MIP) [14] is a framework to find an optimal solution for a linear objective function subject to a set of linear constraints over real and integer variables. We will exploit the close relation between SAT and MIP [15] in order to reduce GenPSAT problems to suitable MIP problems.

As observed in many NP-complete problems [16], GenPSAT also presents a phase transition behaviour. By solving batches of parametrized random GenPSAT problems, we observe the existence of a threshold splitting a phase where almost every GenPSAT problem is satisfiable, from a phase where almost every GenPSAT problem is not satisfiable. During such transition, the problems become much harder to solve [16].

We then recall a generalization of GenPSAT, GGenPSAT, introduced in [17], and whose language allows Boolean combinations of linear inequalities involving probabilities of classical propositional formulas. We use this powerful language to develop applications in information security, namely to formally characterize attackers with side-channel capabilities. Specifically, we study the problem of deciding whether a formula is perfectly masked against such attackers. We focus

on two types of active attackers: *attackers with fault-injection capabilities* which are able to partially (or fully) control the masks being used to protect the leakage of information and *attackers with variable-dependency capabilities* which are able to make two or more, previously independent random variables, dependent.

As the main contribution of this work, throughout Sections 2-4, we develop the theoretical framework that allows the translation between GenPSAT and MIP problems, which then allows the implementation of a provably correct solver for GenPSAT. With the GenPSAT solver in hands, we are able to detect and study the phase transition behaviour of this problem. This work is included in [1]. Then, in Section 5, which partially integrates [17], we present an extension of GenPSAT, GGenPSAT, whose language allows for a powerful modelling of problems in hardware verification and information security. Finally, in Section 6, we use the GGenPSAT probabilistic formalism to characterize attackers with side-channel capabilities and then proceed to formalize the notion of a perfectly masked circuit against such attackers. Surprisingly, when facing a very powerful attacker, this problem actually becomes easier and is shown to be in co-NP.

2. The GenPSAT problem

Let us begin by fixing a set of propositional variables $\mathcal{P} = \{x_1, \ldots, x_n\}$. We define the set of *classical propositional formulas* as

$$\mathrm{L}_{\mathsf{CPL}} ::= \mathcal{P} \mid \neg \mathrm{L}_{\mathsf{CPL}} \mid \mathrm{L}_{\mathsf{CPL}} \wedge \mathrm{L}_{\mathsf{CPL}} \ .$$

Observe that the other logical connectives $\rightarrow, \vee, \leftrightarrow$ can be defined by abbreviation, as usual. A *literal* is either a propositional variable or its negation. A *propositional clause* is a non-empty disjunction of one or more literals. A *propositional formula* is any Boolean combination of propositional variables. We also denote the *size of a classical propositional formula* φ by $|\varphi|$, and is inductively defined as follows: $|x| = 1$ for $x \in \mathcal{P}$; $|c(\varphi_1, \ldots, \varphi_n)| = 1 + |\varphi_1| + \ldots + |\varphi_n|$, where c is an n-ary connective and $\varphi_i \in \mathrm{L}_{\mathsf{CPL}}$.

A *propositional valuation* is a map $v : \mathcal{P} \rightarrow \{0, 1\}$, which is extended to propositional formulas as usual. We say that a set of valuations $\mathcal{V}$ satisfies a propositional formula φ if, for each $v \in \mathcal{V}$, $v(\varphi) = 1$. This notion is extended to sets of propositional formulas as usual. Let $\mathcal{V}^* = \{v_1, \ldots, v_{2^n}\}$ be the set of all valuations defined over variables of $\mathcal{P}$. We define a *probability distribution* π over $\mathcal{V}^*$ as a probability vector of size 2^n.

A *simple probabilistic formula* is an expression of the form $\mathsf{Pr}(c) \boxtimes p$, where c is a clause, $p \in \mathbb{Q}$, $0 \leq p \leq 1$ and $\boxtimes \in \{=, \leq, \geq\}$. We say that a probability distribution π *satisfies* a formula $\mathsf{Pr}(c) \boxtimes p$ if

$$\sum_{i=1}^{2^n} (v_i(c) \cdot \pi_i) \boxtimes p \ .$$

A probability distribution π satisfies a set of simple probabilistic formulas if it satisfies each one of them.

We now recall the PSAT problem [6, 8, 7].

Definition 1 (PSAT problem)**.** *Given a set of propositional variables $\mathcal{P}$ and a set of simple probabilistic formulas $\Sigma = \{\mathsf{Pr}(c_i) \boxtimes p_i \mid 1 \leq i \leq k\}$, the Probabilistic Satisfiability problem (PSAT) consists in determining whether there exists a probability distribution π over $\mathcal{V}^*$ that satisfies Σ.*

The PSAT problem for $\{\mathsf{Pr}(c_i) \boxtimes_i p_i \mid 1 \leq i \leq k\}$ can be formulated algebraically as the problem of finding a solution π for the system of inequalities

$$\begin{cases} V\pi \boxtimes p \\ \sum \pi_i = 1 \\ \pi \geq 0 \end{cases} ,$$

where V is the $k \times 2^n$ matrix such that $V_{ij} = v_j(c_i)$, i.e., $V_{ij} = 1$ iff the j-th valuation satisfies the i-th clause, $p = [p_i]$ is the k vector of all p_i and $\boxtimes = [\boxtimes_i]$ is the k vector of all $\boxtimes_i$.

The SAT problem can be modelled as a PSAT instance where the entries p_i of the probability vector are all identical to 1. The PSAT problem was shown to be NP-complete [8, 13], even when the clauses consist of the disjunction of only two literals, 2-PSAT.

We now extend the notion of simple probabilistic formula to handle linear inequalities involving probabilities of propositional formulas. A *probabilistic formula* is an expression of the form

$$\textstyle\sum_{i=1}^{\ell}(a_i \mathsf{Pr}(\varphi_i)) \bowtie p \ ,$$

where $\varphi_i \in \mathrm{L}_{\mathsf{CPL}}$, $\bowtie \in \{\geq, <, \neq\}$, $\ell \in \mathbb{N}$ and $a_i, p \in \mathbb{Q}$. Observe that formulas with the relational symbols $\leq, >$ can be obtained by abbreviation and formulas with $=$ are obtained as a combination of probabilistic formulas. An *atomic probabilistic formula* is a probabilistic formula where each φ_i is a propositional variable. We say that a probability distribution π *satisfies* a formula $\sum_{i=1}^{\ell}(a_i \mathsf{Pr}(\varphi_i)) \bowtie p$ if

$$\textstyle\sum_{i=1}^{\ell} \left(a_i \left(\sum_{j=1}^{2^n} v_j(\varphi_i) \cdot \pi_j \right) \right) \bowtie p \ .$$

A probability distribution π satisfies a set of probabilistic formulas if it satisfies each one of them.

An *instance* of GenPSAT is a pair (Γ, Σ) where Γ is a set of propositional formulas (also called hard constraints) and Σ is a set of probabilistic formulas (soft constraints). We say that a probability distribution π *satisfies* a GenPSAT instance (Γ, Σ) if it satisfies the set of probabilistic formulas

$$\Xi_{(\Gamma,\Sigma)} = \Sigma \cup \{\mathsf{Pr}(\gamma) = 1 \mid \gamma \in \Gamma\} \ . \tag{1}$$

Definition 2 (GenPSAT problem)**.** *Given a GenPSAT instance (Γ, Σ), the Generalized Probabilistic Satisfiability problem (GenPSAT) consists in determining whether there exists a probability distribution π over $\mathcal{V}^*$ that satisfies (Γ, Σ).*

GenPSAT poses a convenient framework for specifying constraints involving different probabilistic formulas. For instance, one may want to impose that $2\mathsf{Pr}(A) \leq \mathsf{Pr}(B)$ for two propositional formulas A, B. Such requirements may be very useful in specifying properties of interesting systems but they cannot be easily expressed in the PSAT framework. We now showcase GenPSAT's expressiveness by encoding the Monty Hall problem [18].

Example 1. *The Monty Hall problem is a puzzle where we are faced with the choice of picking one of three doors, knowing that a prize is behind one of them. After our initial choice, the game host opens one of the remaining doors provided that the prize is not behind it, and gives us the choice of switching or keeping the initial guess. The question is: which option is more advantageous?*

To model this problem as a GenPSAT *instance, let us define the following propositional variables: P_i holds if the prize is behind door i, X_i holds if our initial choice is door i, H_i holds if the host reveals door i after our initial choice, for $i \in \{1,2,3\}$. Since there are only one door with a prize, one initial choice, and one door revealed by the host, we impose the following restrictions:*

$$\Gamma_1 = \left\{ \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (P_i \wedge \neg P_j \wedge \neg P_k), \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (X_i \wedge \neg X_j \wedge \neg X_k), \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (H_i \wedge \neg H_j \wedge \neg H_k) \right\} .$$

Furthermore, the host cannot open neither the chosen door nor the door with the prize and so we also impose the following constraints:

$$\Gamma_2 = \bigcup_{i\in\{1,2,3\}} \{P_i \rightarrow \neg H_i,\, X_i \rightarrow \neg H_i\} \ .$$

We further assume that the prize has uniform probability of being behind each door and that the initial choice is independent of where the prize is:

$$\Sigma = \textstyle\bigcup_{i,j\in\{1,2,3\}} \left\{ \mathsf{Pr}(P_i) = \frac{1}{3}\,, \quad \mathsf{Pr}(P_i \wedge X_j) = \frac{1}{3}\,\mathsf{Pr}(X_j) \right\}$$

Concerning the question of which is more advantageous, switching or keeping our initial choice, we encode winning by switching *and* winning by keeping*, respectively, as*

$$\mathsf{WbS} : \textstyle\bigwedge_{i=1}^{3} (P_i \leftrightarrow (\neg X_i \wedge \neg H_i))\,, \qquad \mathsf{WbK} : \bigwedge_{i=1}^{3} (P_i \leftrightarrow X_i) \ .$$

We want to the decide whether it is always the case that $\mathsf{Pr}(\mathsf{WbS}) \geq \mathsf{Pr}(\mathsf{WbK})$, which can be checked by testing the satisfiability of the GenPSAT instance

$$(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) < \mathsf{Pr}(\mathsf{WbK})\})\,, \ \textit{where } \Gamma = \Gamma_1 \cup \Gamma_2 \ .$$

As expected, this instance is not satisfiable and the instance $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) \geq \mathsf{Pr}(\mathsf{WbK})\})$ is satisfiable, allowing us to conclude that it is always advantageous to switch our initial option.

We can take this analysis one step further, and show that the probability of winning by switching is $\frac{2}{3}$ by checking that the instance $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) \neq \frac{2}{3}\})$ is unsatisfiable and that the instance $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) = \frac{2}{3}\})$ is satisfiable. All these instances were checked using the tool we implemented, [19]. ◊

Notice that the PSAT problem for Σ can be modelled in GenPSAT by considering the instance $(\emptyset, \Sigma)$.

Given a GenPSAT instance (Γ, Σ), where Γ contains m formulas and Σ is composed of k probabilistic formulas, we follow the lines of Nilsson [6] for a linear algebraic formulation and consider a $(k+m) \times 2^n$ matrix $V = [V_{ij}]$, where for each $i \in \{1, \ldots, k+m\}$ and $j \in \{1, \ldots, 2^n\}$ V_{ij} is defined from the j^{th} valuation v_j and from the i^{th} probabilistic formula $\sum_{u=1}^{\ell} a_u^i \mathsf{Pr}(\varphi_u^i) \bowtie_i p_i$ of $\Xi_{(\Gamma,\Sigma)}$ as follows:

$$V_{ij} = \textstyle\sum_{u=1}^{\ell} a_u^i \cdot v_j(\varphi_u^i) \ .$$

Furthermore, define two vectors of size $k+m$, $p = [p_i]$ and $\bowtie = [\bowtie_i]$. GenPSAT is equivalent to the problem of deciding the existence of a solution π to the system

$$\begin{cases} V\pi \bowtie p \\ \sum \pi_i = 1 \\ \pi \geq 0 \end{cases} \quad . \tag{2}$$

Given a set of probabilistic formulas $\Omega = \left\{ \sum_{u=1}^{\ell} a_u^i \cdot v_j(\varphi_u^i) \bowtie_i p_i \mid 1 \leq i \leq k \right\}$ and a set of valuations $\mathcal{V} = \{v_1, \ldots, v_{k'}\}$, we define the $[\Omega, \mathcal{V}]$-*associated matrix* as the $(k+1) \times k'$ matrix $M_{[\Omega,\mathcal{V}]} = [M_{ij}]$ such that

$$M_{k+1,j} = 1 \text{ for each } 1 \leq j \leq k'$$

and

$$M_{ij} = \sum_{u=1}^{\ell} a_u^i \cdot v_j(\varphi_u^i) \quad \text{for } 1 \leq i \leq k, \ \ 1 \leq j \leq k' \ .$$

Then, we can rewrite system (2) using the $[\Xi_{(\Gamma,\Sigma)}, \mathcal{V}^*]$-associated matrix V as

$$\begin{cases} V\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{3}$$

We now show that this problem is NP-complete. For this purpose, we first present the following lemma.

Lemma 1 ([13, 20])**.** *If a system of ℓ linear inequalities with integer coefficients has a non-negative solution, then it has a non-negative solution with at most ℓ positive entries.*

Theorem 1 ([13])**.** GenPSAT *is* NP*-complete.*

Proof. We begin by showing that GenPSAT is in NP by providing a polynomial sized certificate. Notice that Lemma 1 can be extended to rational coefficients simply by normalizing with the greatest denominator. Applying this result to the system (3) we conclude that there is a $(k+m+1) \times (k+m+1)$ matrix W, composed of columns of V, whose system

$$\begin{cases} W\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{4}$$

has a solution iff the original system (3) has a solution. Furthermore, the obtained solutions from (4) can be mapped to solutions of (3) by inserting zeros in the appropriate positions. Since the solution of this system has $k + m + 1$ elements, it constitutes the NP-certificate for the GenPSAT problem.

Furthermore, given that the PSAT problem can be modelled in GenPSAT, it follows that GenPSAT is NP-complete. □

We say that a GenPSAT instance (Γ, Σ) is in *normal form* if Γ is a set of propositional clauses with 3 literals, i.e., Γ can be seen as a 3CNF formula, and Σ is a set of atomic probabilistic formulas.

Lemma 2. *Given a* GenPSAT *instance* (Γ, Σ) *there exists an instance* (Γ', Σ') *in normal form such that* (Γ, Σ) *is satisfiable iff* (Γ', Σ') *is satisfiable. Moreover,* (Γ', Σ') *is obtained from* (Γ, Σ) *in polynomial time.*

Proof. Let (Γ, Σ) be the GenPSAT instance to be put in normal form. We obtain Σ' by transforming formulas in Σ into atomic probabilistic formulas. For this purpose, let $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(\varphi_i) \bowtie p$ be a formula in Σ and consider the atomic probabilistic formula obtained by replacing (when needed) each formula φ_i by a fresh variable y_i, $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p$. Furthermore, the y_i variable is added to $\mathcal{P}$ and the formula stating the equivalence between y_i and φ_i, $(y_i \leftrightarrow \varphi_i)$, is collected in a set Δ.

We are left with the transformation of the formula

$$\bigwedge_{\gamma \in \Gamma} \gamma \ \wedge \bigwedge_{(y \leftrightarrow c) \in \Delta} (y \leftrightarrow c)$$

into 3-CNF using Tseitin's transformation [21], which can increase linearly the size of the formula and add new variables to $\mathcal{P}$. The final Γ' is the set of conjuncts of the obtained 3-CNF formula. Since Tseitin's transformation preserves satisfiability of formulas, (Γ, Σ) is satisfiable iff (Γ', Σ') is satisfiable. □

3. Reducing GenPSAT to Mixed-Integer Programming

In this section we explore the close relation between satisfaction of propositional formulas and feasability of a set of linear constraints over binary variables (see [15]). With this, we present a reduction of GenPSAT to Mixed-Integer Programming (MIP), similarly to what was done for PSAT [9] and GPSAT [10]. A MIP problem consists in optimizing a linear objective function subject to a set of linear constraints over real and integer variables. MIP was shown to be NP-complete, see [14]. Observe that this translation to MIP also serves as a proof that GenPSAT is in NP.

3.1. Linear Algebraic Formulation for GenPSAT

Lemma 3. *A* GenPSAT *instance in normal form* (Γ, Σ)*, with* $|\Sigma| = k$*, is satisfiable iff there exists a* $(k+1) \times k'$ *matrix* W *of rank* $k' \leq k+1$ *and a set of valuations* $\mathcal{V}_0$ *of size* k' *such that:*

(i) W *is the* $[\Sigma, \mathcal{V}_0]$*-associated matrix*

(ii) $\mathcal{V}_0$ *satisfies* Γ*,*

(iii) considering $p = [p_1, \ldots, p_k, 1]$ *and* $\bowtie = [\bowtie_1, \ldots, \bowtie_k, =]$*, the system*

$$\begin{cases} W\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{5}$$

is satisfiable.

Proof. Let (Γ, Σ) be a satisfiable GenPSAT instance in normal form, with $|\Sigma| = k$ and $|\Gamma| = m$. Then, denoting by V the $[\Xi_{(\Gamma,\Sigma)}, \mathcal{V}^*]$-associated matrix, the system

$$\begin{cases} V\pi \bowtie p \\ \pi \geq 0 \end{cases}$$

has a solution. And so, using Lemma 1, there is a $(k + m + 1) \times \ell$ matrix V^*, where $\ell \leq k + m + 1$, and whose system has a positive solution π^*. Notice that the set of valuations underlying V^* certainly satisfies Γ, as $\pi^*_j > 0$ for each $1 \leq j \leq \ell$.

Let W^* be the matrix constructed from V^* by choosing the first k rows (corresponding to the probabilistic formulas in Σ) and the last row (requiring that the solution sums up to one) of V^*. Still, the corresponding system has a positive solution. Using Lemma 1 once more, we conclude that exists a $(k+1)\times k'$ matrix W, with $k' \leq k+1$, whose system has a positive solution ρ^*. The solution π for (5) is obtained from ρ^* by inserting zeros in the appropriate positions.

Reciprocally, assume that there exists a $(k + 1) \times k'$ matrix W of rank $k' \leq k + 1$ satisfying (i), (ii), (iii), and let π denote the solution for (5). We are looking for a probability distribution π^* satisfying (Γ, Σ). For this purpose, let $\mathcal{V}_0 = \{v_{j_1}, \ldots, v_{j_{k'}}\} \subseteq \mathcal{V}$ denote the set of valuations underlying W according to condition (ii), and define $\pi^* = [\pi^*_i]$, where

$$\pi^*_i = \begin{cases} \pi_i & \text{if } i \in \{j_1, \ldots, j_{k'}\} \\ 0 & \text{otherwise} \end{cases} .$$

The verification that π^* satisfies the GenPSAT instance is now immediate:

- given $\gamma \in \Gamma$, we check that π^* verifies $\mathsf{Pr}(\gamma) = 1$ by observing that the last equality represented on W on (5) leads to $\sum_{s=1}^{k'} \pi_{j_s} = 1$ and so,

$$\sum_{j=1}^{2^n} v_j(\gamma) \cdot \pi^*_j = \sum_{\{j | v_j(\gamma)=1\}} \pi^*_j = \sum_{s=1}^{k'} \pi_{j_s} = 1 \ .$$

- given an atomic probabilistic formula $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p$ in Σ, we recall the definition of π^* and that π is a solution for (5) to conclude that

$$\sum_{i=1}^{\ell} a_i \left(\sum_{j=1}^{2^n} v_j(y_i) \cdot \pi^*_j \right) = \sum_{s=1}^{k'} \left(\sum_{i=1}^{\ell} a_i \cdot v_{j_s}(y_i) \right) \pi_{j_s} \bowtie p \ ,$$

i.e., π^* satisfies the formulas in Σ. □

3.2. Translation to MIP

Regarding Lemma 3, given a GenPSAT instance (Γ, Σ) in normal form, with $|\Sigma| = k$ and $|\Gamma| = m$, our goal is now to describe a procedure that encodes the problem of finding a set of valuations $\mathcal{V}_0$ and a probability distribution π in the conditions (i),(ii),(iii), as a MIP problem. We dub this procedure GenToMIP.

Let us denote by $H = [h_{ij}]$ the (still unknown) matrix of size $n \times k'$ whose columns represent the valuations in $\mathcal{V}_0$ evaluated on each propositional variable of $\mathcal{P}$, i.e., $h_{ij} = v_j(x_i)$ for each $1 \leq i \leq n$ and $1 \leq j \leq k'$. Let $\alpha_1, \ldots, \alpha_n$ represent the probability of the propositional variables $x_1, \ldots, x_n$, respectively, and following the reasoning of [9, 10] we model the non-linear constraint $\sum_{j=1}^{k'} h_{ij} \cdot \pi_j = \alpha_i$ as a linear inequality

$$\sum_{j=1}^{k'} b_{ij} = \alpha_i \ , \tag{val1}$$

by introducing the extra variables b_{ij} which are subject to the appropriate constraints, namely forcing b_{ij} to be zero whenever $h_{ij} = 0$, and ensuring that $b_{ij} = \pi_j$ whenever $h_{ij} = 1$, i.e.,

$$0 \leq b_{ij} \leq h_{ij} \text{ and } h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j \ . \tag{val2}$$

We ensure that π represents a probability distribution by imposing that

$$\sum_{j=1}^{k'} \pi_j = 1 \ . \tag{sums1}$$

Still, as each valuation of $\mathcal{V}_0$ satisfies Γ, given a clause $\left(\bigvee_{r=1}^{w} x_{i_r}\right) \vee \left(\bigvee_{s=1}^{w'} \neg\, x_{i'_s}\right)$ of Γ, we generate a linear inequality for each valuation $1 \leq j \leq k'$,

$$\left(\sum_{r=1}^{w} h_{i_r,\, j}\right) + \left(\sum_{s=1}^{w'} (1 - h_{i'_s\, j})\right) \geq 1. \tag{gamma}$$

Notice that, if we have a total of m clauses in Γ, we generate $m \times k'$ such inequalities.

In order to verify the satisfiability of probabilistic formulas in the MIP framework, consider an atomic probabilistic formula $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p$ in Σ. Since $\bowtie$ can either be the relational symbol $\geq$, $<$ or $\neq$, we can easily encode the first kind of inequalities as a MIP linear constraint, but should be careful when dealing with the remaining relational symbols.

For atomic probabilistic formulas of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \geq p$, we generate the linear inequality

$$\sum_{i=1}^{\ell} a_i \cdot \alpha_i \geq p \ . \tag{prob$_{\geq}$}$$

In the case where $\bowtie$ is a strict inequality $<$, we use a specific variable introduced into the MIP problem, say ε, to fix the objective function as the maximization of ε,

$$\text{maximize } \varepsilon \tag{obj}$$

and further introduce the linear constraint

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) + \varepsilon \leq p \ . \qquad (\mathsf{prob}_<)$$

For atomic probabilistic formulas φ of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \neq p$, i.e.

$$\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) - p \neq 0, \qquad (6)$$

we force the left hand side to be either strictly greater or strictly less than zero,

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) - p < 0 \qquad \text{or} \qquad \sum_{i=1}^{\ell}(a_i \cdot \alpha_i) - p > 0 \ .$$

Even though these are linear constraints, the problem would explode if we treated the disjunction. In this sense, notice that, denoting by C a sufficiently large number, say $C = 1 + |p| + \sum_{i=1}^{\ell} |a_i|$, the inequality (6) holds if and only if there exists a fresh binary variable z_φ such that the following two strict inequalities hold simultaneously:

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) - p < C \cdot z_\varphi \qquad \text{and} \qquad -\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) + p < C - C \cdot z_\varphi \ .$$

Then, we are left with two strict inequalities, thus reducing this analysis to a previous case, from which we obtain the constraints

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) - p + \varepsilon \ \leq \ C \cdot z_\varphi \qquad \text{and} \qquad -\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) + p + \varepsilon \ \leq \ C - C \cdot z_\varphi \ . \qquad (\mathsf{prob}_\neq)$$

Denoting by $k_\geq$, $k_<$, $k_\neq$ the number of probabilistic formulas in Σ when $\bowtie$ coincides with $\geq$, $<$, $\neq$, respectively, so far we have introduced: n constraints ($\mathsf{val1}$), $4 \times n \times k'$ constraints ($\mathsf{val2}$), 1 constraint ($\mathsf{sums1}$), $m \times k'$ constraints (gamma), $k_\geq$ constraints ($\mathsf{prob}_\geq$), $k_<$ constraints ($\mathsf{prob}_<$), $2 \times k_\neq$ constraints ($\mathsf{prob}_\neq$). Hence, we have $\mathcal{O}(n + n \times k' + m \times k' + k)$ inequalities over $n \times k'$ binary variables h_{ij}, $n \times k'$ real variables b_{ij}, n real variables $0 \leq \alpha_i \leq 1$, $k_\neq$ binary variables z_φ, a real variable $\varepsilon \geq 0$ and k' real variables $\pi_j \geq 0$. Because of this, the GenToMIP translation is polynomial.

Proposition 1. *The* GenToMIP *procedure transforms a* GenPSAT *instance in normal form* (Γ, Σ) *into a* MIP *problem whose size is polynomial on the size of* (Γ, Σ).

We now need to show that the existence of a set of valuations $\mathcal{V}_0$ and a probability distribution π in the conditions (i),(ii),(iii) of Lemma 3 is equivalent to the feasibility of the MIP problem obtained through GenToMIP with an optimal value $\varepsilon > 0$ (when applicable).

This procedure is presented in Algorithm 1, which given a GenPSAT instance, translates it into a MIP problem and then solves the latter appropriately. For that, let us assume that we initialize an empty MIP problem and consider the following auxiliary procedures:

- add_const introduces a linear constraint into the MIP problem,

- set_obj defines the objective function (either as a maximization or as a minimization) when it was previously not defined,
- fresh declares a fresh binary variable into the MIP problem,
- mip_sat returns True or False depending on whether the problem is feasible (and achieves an optimal solution) or not,
- mip_objvalue returns the objective value, if an objective function was set.

Algorithm 1 GenPSAT solver based on MIP

procedure GenPSAT(props $\{x_i\}_{i=1}^n$, form Γ, probform Σ)
  declare: binary variables: h_{ij} for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,k'\}$
    $[0,1]$-variables: α_i, π_j, b_{ij} for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,k'\}$
    real variable: ε
  for $j = 1$ to k' **do**
    for each $(\bigvee_r x_r) \vee (\bigvee_s \neg x_s)$ in Γ **do**
      add_const($\sum_r h_{rj} + \sum_s (1 - h_{sj}) \geq 1$) ▷ (gamma)
  for $i = 1$ to n **do**
    add_const($\sum_j b_{ij} = \alpha_i$) ▷ (val1)
    for $j = 1$ to k' **do**
      add_const($0 \leq b_{ij} \leq h_{ij}$) ▷ (val2)
      add_const($h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j$) ▷ (val2)
  $aux \leftarrow 0$
  for each $\sum a_i \cdot \mathsf{Pr}(x_i) \bowtie q$ in Σ **do**
    switch($\bowtie$)
      case " $\geq$ " :
        add_const($\sum a_i \cdot \alpha_i \geq q$) ▷ (prob$_\geq$)
      case " $<$ " :
        $aux \leftarrow 1$
        set_obj(max ε) ▷ (obj)
        add_const($\sum a_i \cdot \alpha_i + \varepsilon \leq q$) ▷ (prob$_<$)
      case " $\neq$ " :
        $aux \leftarrow 1$
        $z \leftarrow$ fresh() ▷ z is a fresh binary variable
        $C \leftarrow 1 + |q| + \sum |a_i|$
        set_obj(max ε) ▷ (obj)
        add_const($\sum a_i \cdot \alpha_i - C \cdot z - \varepsilon \geq q - C$) ▷ (prob$_\neq$)
        add_const($\sum a_i \cdot \alpha_i - C \cdot z + \varepsilon \leq q$) ▷ (prob$_\neq$)
  add_const($\sum \pi_i = 1$) ▷ (sums1)
  if mip_sat() **then**
    if ($aux == 0$) or ($aux == 1$ and mip_objvalue() > 0) **then**
      return Sat
  return Unsat

Proposition 2. *A* GenPSAT *instance in normal form* (Γ, Σ) *is satisfiable iff Algorithm 1 returns* Sat.

Proof. Let (Γ, Σ) be a satisfiable GenPSAT instance in normal form, and also $\mathcal{V}_0 = \{v_1, \ldots, v_{k'}\}$ and $\rho = [\rho_i]$ represent a set of valuations and a probability distribution given by Lemma 3 which satisfy conditions (i)-(iii). Then, consider the following values and afterwards let us check that they constitute an optimal solution for the MIP problem constructed at Algorithm 1: for each $1 \leq i \leq n$ and $1 \leq j \leq k'$, let

$$\begin{aligned} h^*_{ij} &= v_j(x_i), \\ b^*_{ij} &= h^*_{ij} \cdot \rho_j, \\ \pi^*_j &= \rho_j, \\ \alpha^*_i &= \sum_{\{j | v_j(x_i)=1\}} \rho_j, \\ \varepsilon^* &= \mathsf{min}\ \Delta, \end{aligned}$$

where $\Delta = \{q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) \mid (\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) < q) \in \Sigma\} \cup$
$\cup \{C \cdot z^*_\varphi + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) \mid \varphi \in \Sigma \text{ is of the form } \sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q\} \cup$
$\cup \{C - C \cdot z^*_\varphi - q + \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) \mid \varphi \in \Sigma \text{ is of the form } \sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q\}$,
and, for each atomic probabilistic formula $\varphi \in \Sigma$ of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q$,

$$z^*_\varphi = \begin{cases} 0, & \text{if } \sum_{i=1}^{\ell} a_i \cdot \alpha^*_i < q \\ 1, & \text{if } \sum_{i=1}^{\ell} a_i \cdot \alpha^*_i > q \end{cases} .$$

Now let us check that each linear constraint introduced into the MIP problem at Algorithm 1 is satisfied.

(gamma) $\{h^*_{ij}\}$ satisfy the constraints modelling Γ since each $v \in \mathcal{V}_0$ satisfies Γ.

(val1) By definition of $\{b^*_{ij}\}$ and $\{h^*_{ij}\}$, we actually have

$$\sum_{j=1}^{k'} b^*_{ij} = \sum_{j=1}^{k'} h^*_{ij} \cdot \rho_j = \sum_{j=1}^{k'} v_j(x_i) \cdot \rho_j = \sum_{\{j | v_j(x_i)=1\}} \rho_j = \alpha^*_i \ .$$

(val2) Since $0 \leq v_j(x_i) \leq 1$ and $0 \leq \rho_j \leq 1$ we immediately have $0 \leq b^*_{ij} \leq h^*_{ij}$.

For the other inequality, recall that $h^*_{ij} = v_j(x_i)$ and that $\pi^*_j = \rho_j$ and note that:

- if $h^*_{ij} = 0$ then $b^*_{ij} = 0$ and, since $\pi^*_j \leq 1$, it follows that $\pi^*_j - 1 \leq b^*_{ij} \leq \pi^*_j$, i.e., $h^*_{ij} - 1 + \pi^*_j \leq b^*_{ij} \leq \pi^*_j$
- if $h^*_{ij} = 1$ then $b^*_{ij} = \pi^*_j$ and so $\pi^*_j \leq b^*_{ij} \leq \pi^*_j$, i.e., $h^*_{ij} - 1 + \pi^*_j \leq b^*_{ij} \leq \pi^*_j$

(sums1) Since $\pi^*_j = \rho_j$, we immediately conclude that $\sum_{j=1}^{k'} \pi^*_j = 1$.

To check that the probabilistic formulas are satisfiable, just note that, given a probabilistic formula $(\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \bowtie q) \in \Sigma$,

$$\sum_{i=1}^{\ell} a_i \cdot \alpha_i^* = \sum_{i=1}^{\ell} a_i \left(\sum_{\{j | v_j(x_i)=1\}} \rho_j \right) = \sum_{i=1}^{\ell} a_i \left(\sum_{j=1}^{2^n} v_j(x_i) \cdot \rho_j \right).$$

($\mathsf{prob}_{\geq}$) Let $(\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \geq q) \in \Sigma$ and notice that, since ρ satisfies conditions (i), (ii), (iii), in particular it satisfies all the probabilistic formulas in Σ, and so $\sum_{i=1}^{\ell} a_i \left(\sum_{j=1}^{2^n} v_j(x_i) \cdot \rho_j \right) \geq q$, which implies that $\sum_{i=1}^{\ell} a_i \cdot \alpha_i^* \geq q$.

($\mathsf{prob}_{<}$) Now, let $(\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) < q) \in \Sigma$ and notice that, in a reasoning very similar to the previous one, we can conclude that $\sum_{i=1}^{\ell} a_i \cdot \alpha_i^* < q$, i.e.

$$q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) > 0. \tag{7}$$

But we should also note that, since $\varepsilon^* = \mathsf{min}\ \Delta$, then $\varepsilon^* \leq q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*)$, and so we obtain

$$\sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) + \varepsilon^* \leq \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) + q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) = q \ .$$

($\mathsf{prob}_{\neq}$) Finally, let us consider an atomic probabilistic formula $\varphi \in \Sigma$ of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q$, and recall once more that since ρ satisfies each probabilistic formula of Σ, we have $\sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) \neq q$, in other words, either $q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) > 0$ or $q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) < 0$. Recall the constant C defined as $C = 1 + |q| + \sum_{i=1}^{\ell} |a_i|$ and the definition of z_φ^* and notice that both

$$C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) > 0 \tag{8}$$

and

$$C - C \cdot z_\varphi^* - q + \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) > 0 \tag{9}$$

are verified in either of the above cases. Also note that by definition of ε^*, $\varepsilon^* \leq C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*)$ and $\varepsilon^* \leq C - C \cdot z_\varphi^* - q + \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*)$. Hence, we now analyze each of the previous cases:

- if $q > \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*)$, then $z_\varphi^* = 0$ and it follows that
  $\sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) - C \cdot z_\varphi^* - \varepsilon^* \geq \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) - C + C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) = q - C,$

  and further,
  $\sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) - C \cdot z_\varphi^* + \varepsilon^* \leq \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) + C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) = q.$

- if $q < \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*)$, then $z_\varphi^* = 1$ and it follows that

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) - C \cdot z_\varphi^* - \varepsilon^* \geq \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) - C - (C - C \cdot z_\varphi^* - q + \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*)) = q - C,$$

and further,

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) - C \cdot z_\varphi^* + \varepsilon^* \leq \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) - C + C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) = q.$$

To finish the direct implication, notice that $\varepsilon^* > 0$ as a consequence of (7), (8) and (9), and it takes the maximum possible value since otherwise, let φ_Δ be the formula in Σ which has the minimum value in Δ. Then, if there was a solution with greater objective value it would violate the constraint ($\mathsf{prob}_{\bowtie}$) for φ_Δ.

Reciprocally, assume that Algorithm 1 returned Sat, and let us denote by h_{ij}^*, α_i^*, ε^* and π_j^* the (optimal) solution for the variables h_{ij}, α_i, ε and π_j, for each $1 \leq i \leq n$, $1 \leq j \leq k'$ respectively.

Consider the set of valuations $\mathcal{V}_0 = \{v_1, \ldots, v_{k'}\}$ where, for each propositional variable $x_i \in \mathcal{P}$, $v_j(x_i) = h_{ij}^*$. Due to constraints (gamma) it is immediate to conclude that each valuation satisfies Γ. Then, let the probability distribution π be defined over the set of valuations as the 2^n vector $\pi = [\rho_j]$ where $\rho_j = \pi_j^*$ for $1 \leq j \leq k'$ and $\rho_j = 0$ for $k' < j \leq 2^n$. Note that (sums1) implies that π is a probability vector. The third condition described in Lemma 3 is deduced by simple inspection of the linear constraints ($\mathsf{prob}_{\geq}$), ($\mathsf{prob}_{<}$), ($\mathsf{prob}_{\neq}$) and (sums1), by definition of the matrix associated to Σ over $\mathcal{V}_0$ and recalling that the optimal value ε^* is such that $\varepsilon^* > 0$. □

As a corollary of the previous propositions, we obtain the following result.

Theorem 2. *The* GenToMIP *algorithm is a correct polynomial time translation of* GenPSAT *to a* MIP *problem.*

4. Phase Transition

Phase transition is a phenomenon that marks a hardness shift in the solution of instances of a problem. This behaviour was observed in many NP-complete problems [16], among which we highlight 3-SAT [22] and PSAT [7, 11].

In this section, we study the GenPSAT phase transition, through an implementation of Algorithm 1 and tests comprised of batteries of random instances. For this, we measure the proportion of satisfiable instances as well as the average time the solver spent to solve them. The software was written in Java, and we used Gurobi [23], version 6.5.0, to solve the MIP problem. The machine used for the tests was a Mac Pro at 3,33 GHz 6-Core Intel Xeon with 6 GB of memory. Our implementation is available in [19].

It was noted that, in random 3-SAT instances [22] there is a clear stage where the instances are almost surely satisfiable and one where they are almost surely not satisfiable. This phenomenon is characterized by the existence of a threshold value for the ratio m/n, where m is the number of clauses, and n is the number of variables, for which: for smaller values of the ratio, the SAT

instances are almost certainly satisfiable and easily solved, whereas instances with larger ratio values are almost certainly unsatisfiable and also easily solved. However, with values of the ratio very closed to this threshold, the instances are, on average, very hard to solve and there is no certainty on whether the problem is satisfiable or not. As we have already noted, any 3-SAT problem can be seen as a GenPSAT instance. We tested our GenPSAT solver with random instances of 3-SAT, and observed that a phase transition occurs when the ratio m/n is about 4.3, in accordance with [22], see Figure 1.

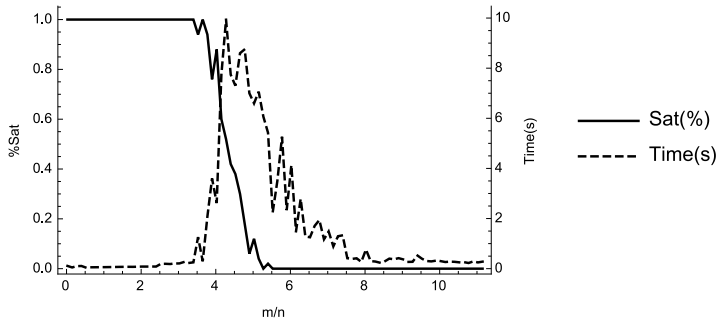


Figure 1: Phase transition for SAT seen as a GenPSAT instance, with $n = 20$.

A deeper analysis of the probabilistic satisfiability problem PSAT [7, 11] has shown the presence of a phase transition behaviour for PSAT for a ratio m/n, where m is the number of clauses and n is the number of variables. We tested random PSAT instances with the number of probabilistic formulas $k = 2$, $n = 15$ and m ranging from 1 to 105 in steps of 2. For each value of m, we generated 100 PSAT instances. The obtained results are presented in Figure 2.

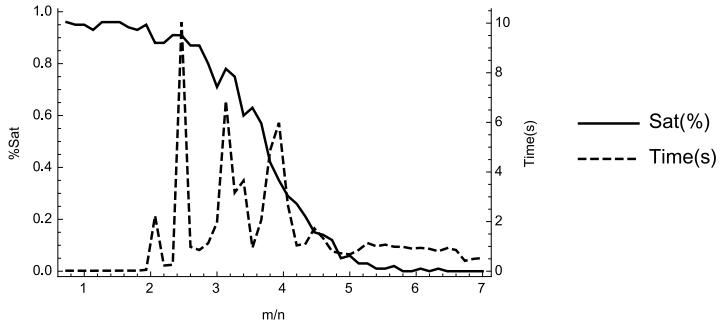


Figure 2: PSAT phase transition seen as a GenPSAT instance, with $n = 15$ and $k = 2$.

We highlight that the analysis of the existence of a phase transition with variation on k (instead of a variation on m) is essential for a deep understanding of the phase transition of the probabilistic satisfiability problem (instead of the phase transition of the satisfiability problem for propositional formulas in the presence of probabilistic formulas). For this purpose, we tested random PSAT instances with $n = 30$, $m = 40$ and k ranging from 1 to 25, and also observed a

phase transition with respect to k/n based on 100 instances for each value of k, see Figure 3.

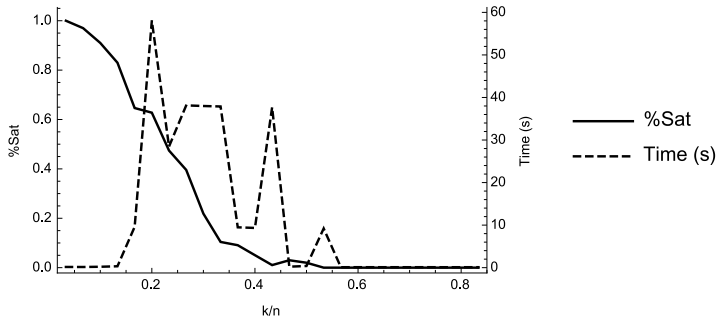


Figure 3: PSAT phase transition seen as a GenPSAT instance, with $n = 30$ and $m = 40$.

In [10], this phase transition analysis was performed on a generalization of the probabilistic satisfiability problem, GPSAT, which consists in Boolean combinations of simple probabilistic formulas.

In what concerns GenPSAT, notice that a randomly sampled probabilistic formula can easily be inconsistent by itself, e.g., when it implies one of the probabilities is greater than 1. Because of this, the sampling of the coefficients was performed so that this case does not occur.

GenPSAT gives us a wider scope of ratios to study the phase transition behaviour. Due to its generalized nature, we have four dimensions to explore: the number of variables n, the number of clauses m, the number of probabilistic formulas k and the maximum size of the linear combination into the probabilistic formulas ℓ. Here, we analyze the presence of phase transition for k/n and m/n.

By performing random tests, we observe the presence of a phase transition for the ratio of k/n with a very short stage of satisfiable formulas. This is explained since a GenPSAT instance is more likely to be unsatisfiable. Figure 4 represents the phase transition for random GenPSAT instances with $n = 20$, $m = 10$ and k ranging from 1 to 100 in steps of 2. We generated 100 instances for each value of k.

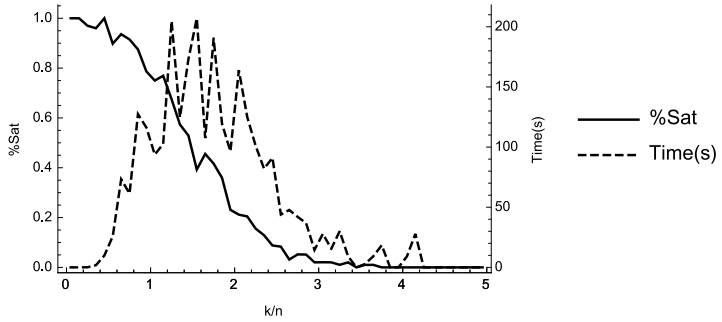


Figure 4: Phase transition for random GenPSAT instances, with $n = 20$ and $m = 10$.

On the other hand, when the parameters n and k are fixed, we are also able to detect a phase transition. Figure 5 represents the result of testing random GenPSAT instances with $n = 15$, $k = 2$ and m ranging from 1 to 105 in steps of 2. For each value of m we generated 100 GenPSAT instances.

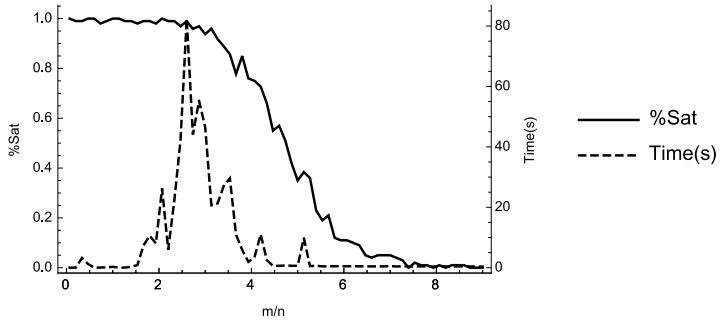


Figure 5: Phase transition for random GenPSAT instances, with $n = 15$ and $k = 2$.

5. Extending GenPSAT

An extension of GenPSAT to encompass Boolean combinations of the GenPSAT probabilistic formulas is a natural path to follow. Even more so, since this language coincides precisely with the language of the probabilistic logic of Fagin et al. presented in [13].

We now present this extension, called GGenPSAT, introduced in [17], as well as some example applications of this formalism to information security which have been verified using the solver [24]. We should note that this solver, which extends the scope of application to GGenPSAT, uses a reduction to SMT contrarily to the solver [19] whose reduction to MIP was presented in Section 3 and in [1]. More details on this particular reduction, the technology used for the solver as well as a phase transition study for this problem can be found in [17, 24].

Having fixed a set of propositional variables $\mathcal{P} = \{x_1, \ldots, x_n\}$ recall that the set of *classical propositional formulas* is defined by

$$\mathrm{L_{CPL}} ::= \mathcal{P} \mid \neg \mathrm{L_{CPL}} \mid \mathrm{L_{CPL}} \wedge \mathrm{L_{CPL}} \ .$$

We recall from [13] the set of *probabilistic atoms* (used herein to define Boolean probabilistic formulas) composed by linear inequalities of probabilities of propositional formulas with rational coefficients:

$$\mathsf{PAt} ::= \mathbb{Q} \cdot \mathsf{Pr}(\mathrm{L_{CPL}}) + \ldots + \mathbb{Q} \cdot \mathsf{Pr}(\mathrm{L_{CPL}}) \geq \mathbb{Q} \ .$$

The set of *Boolean probabilistic formulas* is defined as a Boolean combination of probabilistic atoms as follows:

$$\mathsf{Prob} ::= \mathsf{PAt} \mid \neg \mathsf{Prob} \mid \mathsf{Prob} \wedge \mathsf{Prob} \ .$$

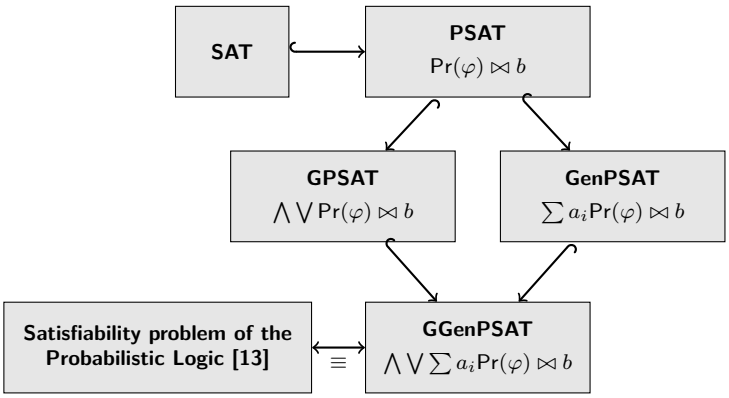


Figure 6: Inclusion diagram of several fragments of the probabilistic logic

Again, the other relational symbols $\{<, >, \leq, =, \neq\}$ can be defined by abbreviation, as well as the logical connectives $\rightarrow, \vee, \leftrightarrow$. For the sake of simplicity, we will denote Boolean probabilistic formulas by probabilistic formulas.

To interpret probabilistic formulas, we consider a probability distribution π over $\mathcal{V}^*$. The satisfaction relation is inductively defined as:

- $\pi \Vdash q_1 \cdot \mathsf{Pr}(\varphi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q$ iff $\sum_{i=1}^{\ell} \left(q_i \left(\sum_{j=1}^{2^n} v_j(\varphi_i) \cdot \pi_j \right) \right) \geq q$;
- $\pi \Vdash \neg\delta$ iff $\pi \nVdash \delta$;
- $\pi \Vdash \delta_1 \wedge \delta_2$ iff $\pi \Vdash \delta_1$ and $\pi \Vdash \delta_2$,

where $\delta, \delta_1, \delta_2 \in \mathsf{Prob}$, $q, q_i \in \mathbb{Q}$ and $\varphi_i \in \mathrm{L}_{\mathsf{CPL}}$ where $i \in \{1, \ldots, \ell\}$. A probability distribution π *satisfies* $\delta \in \mathsf{Prob}$ if $\pi \Vdash \delta$ and satisfies a set of probabilistic formulas if it satisfies each one of them.

In the first sections, which contain [1], probabilistic satisfiability was extended to handle linear inequalities involving assignments of values to propositional formulas. We now aim to extend the GenPSAT problem in order to cope with Boolean combinations of probabilistic atoms.

An *instance* of GGenPSAT is a pair (Γ, Ψ) where Γ is a set of classical propositional formulas (also called hard constraints) and Ψ is a set of probabilistic formulas (soft constraints). We say that a probability distribution π *satisfies* a GGenPSAT instance (Γ, Ψ) if it satisfies the set of probabilistic formulas

$$\Xi_{(\Gamma,\Psi)} = \Psi \cup \{\mathsf{Pr}(\gamma) = 1 \mid \gamma \in \Gamma\} \ . \tag{10}$$

Despite the similarities between a GenPSAT and a GGenPSAT instance, the latter allows more expressive probabilistic formulas by allowing Boolean combinations of probabilistic atoms.

Definition 3 (GGenPSAT problem [17])**.** *Given a* GGenPSAT *instance* (Γ, Ψ)*, the Classical Generalized Probabilistic Satisfiability problem consists in determining if there exists a probability distribution* π *over* $\mathcal{V}^*$ *that satisfies* (Γ, Ψ)*.*

GGenPSAT extends the scope of PSAT and GenPSAT by dealing with Boolean combinations of probabilistic formulas. In this way, we are not only able to assign values to probabilities of propositional variables or linear inequalities involving them, but also able to express powerful probabilistic assertions. For instance, we can easily model and reason about a framework where a variable x is either true or false with probability 1 but we do not know which is the case:

$$\mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1 \ .$$

Notice that GGenPSAT has been studied in the context of the decision problem for the probabilistic logic introduced by Fagin, Halpern and Megiddo in [13]. Hence, the computational complexity of this problem is known and addressed in the following theorem.

Theorem 3 ([13])**.** GGenPSAT *is* NP*-complete.*

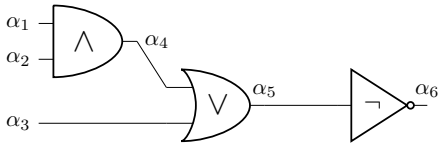
5.1. Applications

We will now showcase novel applications of the GGenPSAT formalism. In particular we will model examples from hardware verification and side-channel attacks in the GGenPSAT mindset. The problem specifications that can be run in the tool, can be found in [24].

5.1.1. Hardware verification

Formal verification in general has greatly impacted hardware verification, allowing the minimization of circuit sizes, bug finding and general design problems. With a probabilistic formalism, we are also able to model unreliable circuits, as well as determine if a certain circuit satisfies a safety guarantee, e.g., work 96% of the time as expected.

The first example is from [25] and studies the implementation of a circuit for an 2-1 AND-OR-INVERTER (AOI21). An AOI21 is the function on 3 bits given by $\mathsf{AOI21}(\alpha_1, \alpha_2, \alpha_3) = \neg((\alpha_1 \wedge \alpha_2) \vee \alpha_3)$. A circuit for this function is for instance the following:



whose implementation is given by the formula

$$\mathsf{Impl} \triangleq (\alpha_4 \leftrightarrow \alpha_1 \wedge \alpha_2) \wedge (\alpha_5 \leftrightarrow \alpha_3 \vee \alpha_4) \wedge (\alpha_6 \leftrightarrow \neg\alpha_5) \ .$$

The validity of this implementation can be easily checked by verifying the unsatisfiability of

$$\mathsf{Impl} \wedge \neg(\alpha_6 \leftrightarrow \mathsf{AOI21}(\alpha_1, \alpha_2, \alpha_3)) \ .$$

More interestingly, we might want to study the reliability of the whole circuit, depending on the reliability of each individual gate. For instance, suppose that the $\wedge$-gate works as expected at least 97% of the time, the $\vee$-gate works at least 99% of the time and the $\neg$-gate always produces the expected value. This description of the circuit can be formalized in GGenPSAT as

$$\widetilde{\mathsf{Impl}} \triangleq \mathsf{Pr}(\alpha_4 \leftrightarrow \alpha_1 \wedge \alpha_2) \geq 0.97 \wedge \mathsf{Pr}(\alpha_5 \leftrightarrow \alpha_3 \vee \alpha_4) \geq 0.99 \wedge \mathsf{Pr}(\alpha_6 \leftrightarrow \neg\alpha_5) = 1 \ .$$

Suppose we now would like to guarantee that this implementation in fact computes the AOI21 function at least 96% of the time, under the previous assumptions on the faulty gates. This is modelled by the following formula

$$\widetilde{\mathsf{Spec}} \triangleq \mathsf{Pr}(\alpha_6 \leftrightarrow \neg((\alpha_1 \wedge \alpha_2) \vee \alpha_3)) \geq 0.96 \ .$$

Hence, to guarantee that under the reliability of the gates we reach this performance, we need to check the satisfiability of

$$\widetilde{\mathsf{Impl}} \wedge \neg\widetilde{\mathsf{Spec}} \ .$$

If it is not satisfiable, we are guaranteed that indeed this circuit works as an AOI21 at least 96% of the time. This can be formally verified in the GGenPSAT solver [24] and is indeed an unsatisfiable set of formulas. Furthermore, notice how this example does not make full use of the expressiveness of the whole probabilistic logic. In particular, we do not make use of linear combinations of probabilistic terms. To showcase this, we present an example on applications to side-channel analysis.

5.1.2. Boolean masking

Consider a circuit with 3 Boolean inputs which computes the function $\varphi(k, r_1, r_2) = k \oplus (r_1 \oplus r_2)$, where k is a secret that is to be masked using the exclusive or, $\oplus$, of two independent $\mathsf{Bernoulli}(\frac{1}{2})$ random Boolean variables r_1, r_2. Our goal is to determine if this mask actually works, i.e., whether it reveals the value of k if we sample the value of $\varphi(k, r_1, r_2)$ enough times. For this, we consider the probability of the circuit returning 1 depending on the value of k. If this probability differs with k, we can sample the circuit to determine k.

To model this problem, we need to find two keys k, k' such that the probability of the formula $\varphi(k, r_1, r_2)$ differs from the probability of $\varphi(k', r_1, r_2)$ and thus forcing that $\mathsf{Pr}(k \oplus (r_1 \oplus r_2)) \neq \mathsf{Pr}(k' \oplus (r_1 \oplus r_2))$. To define each key as fixed but unknown, we enforce that $\mathsf{Pr}(k) = 0 \vee \mathsf{Pr}(k) = 1$ and $\mathsf{Pr}(k') = 0 \vee \mathsf{Pr}(k') = 1$. Modelling uniform random variables is simple in GGenPSAT, $\mathsf{Pr}(r_i) = \frac{1}{2}$ for $i = 1, 2$. Regarding the independence of r_1 and r_2 we should impose that $\mathsf{Pr}(r_1 \wedge r_2) = \mathsf{Pr}(r_1)\mathsf{Pr}(r_2)$ which is not possible since the language does not have products of probabilistic terms. However, when the probability of each random variable $\mathsf{Pr}(r_i)$ is known this can always be expressed, despite leading to an exponential number of formulas [26]. Thus, independence can be modelled as $\mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{4}$. Finally, the whole problem can be described in GGenPSAT

as the following set of assertions:

$$\begin{cases} \mathsf{Pr}(k \oplus (r_1 \oplus r_2)) \neq \mathsf{Pr}(k' \oplus (r_1 \oplus r_2)) \\ \mathsf{Pr}(k) = 0 \vee \mathsf{Pr}(k) = 1 \\ \mathsf{Pr}(k') = 0 \vee \mathsf{Pr}(k') = 1 \\ \mathsf{Pr}(r_i) = \frac{1}{2} \qquad \text{for } i = 1, 2 \\ \mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{4} \end{cases}$$

Encoding this set of assertions in the solver [24] we obtain that the set of assertions is unsatisfiable. This means that indeed, this formula is secure under $\mathsf{Bernoulli}(\frac{1}{2})$ and independent random masks, since there is no model in which the weight of the formula $k \oplus (r_1 \oplus r_2)$ depends on the value of k. However, if we drop the independence restriction, $\mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{4}$, we obtain a satisfiable instance and thus, the circuit would leak information about the secret key if used with non-independent random masks, e.g. $r_1 = r_2$ implies $\varphi(k, r_1, r_2) = k$.

6. A Probabilistic Formalization of Attackers with Side-Channel Capabilities

Even cryptographic protocols which are based on mathematically hard to solve problems can be easily exploited when enough "physical" information is leaked to the outside by the system implementing it – this was the lesson taught by the seminal paper on the power of side-channel attacks, [27]. Side-channel attacks happen when an attacker is able to obtain supposedly private data through information that the system leaks via physical channels such as timing data [27], power consumption [28], electromagnetic radiation [29, 30] to name some impactful side channels. These channels can also be used to exfiltrate data in a covert manner, where an attacker is able to encode data in these channels. With such huge attack surface, this area has not stopped developing and has showed that commonly used implementations of cryptographical protocols and primitives are not safe against these attacks, RSA, DSS and Diffie-Hellman [27], elliptic curve implementations in the GnuPG's Libgcrypt [31] as well as symmetric encryption schemes such as DES [28] and AES [32]. These attacks even extend to quantum protocols such as quantum key distribution [33], which in theory are physically secure.

Despite the security guarantees of the protocols, often enough, their security proofs do not usually take into account the information leaks the system may have through physical properties. In the traditional cryptography view of the world, an attacker only observes the public part of the protocol. For instance, in a private-key encryption scenario Eve, the attacker, would only have access to the encryption of the message exchanged between Alice and Bob, see Figure 7.

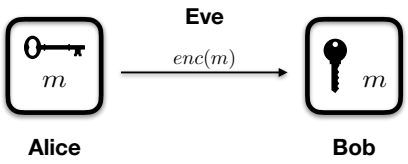


Figure 7: The traditional cryptography view of the world.

However, in the real-world, the encryption process takes time, the machine in which the encryption is being made consumes power, emits heat, electromagnetic radiation and sound, as depicted in Figure 8. Unless defensive measures are taken into account more than one of these channels of physical information about the system may be available to an attacker. Furthermore, the attacker can actively force information leaks by injecting or forcing faults in the system. These fault attacks, introduced in [34], can often lead to full secret recovery, even on standard ciphers such as AES [35].

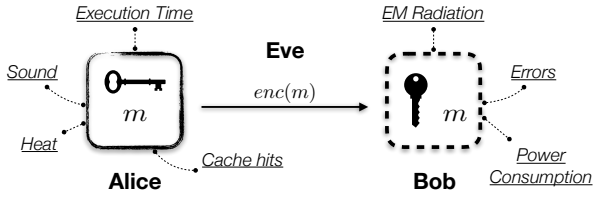
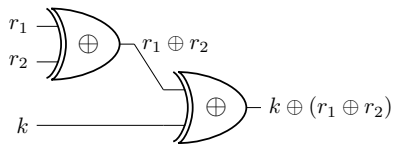


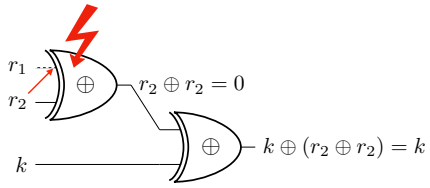
Figure 8: The side-channel view of the world.

There are some usual approaches to thwart these attacks: on one hand there can be a physical shielding of the device running the cryptographic functions, trying to prevent unwanted leakage of information; on the other hand, there can be a logical shielding of the secrets, e.g., by means of a random mask which is applied to values during the execution of algorithm [36, 37]. In this section, we study the latter case, that is, how to develop algorithms with a provably secure implementation. We also only focus on power side-channel attacks – for instance timing side-channel attacks have other countermeasures such as constant-time code, which are out of the scope of this work.

In this section, we use GGenPSAT to formalize the notion of perfect masking of a Boolean formula introduced in [38]. Furthermore, we generalize the notion of perfect masking to encompass an active attacker interfering with the system, and consider the problem of deciding if a formula is perfectly masked under the attack of such an adversary (or a general family of attackers). For this, we consider two generic families of active attackers: *attackers with fault-injection capabilities* which are able to partially (or fully) control the masks being used to protect the leakage of information and *attackers with variable-dependency capabilities* which are able to make two or more, previously independent random variables, dependent. The latter attacker can seem devoid of practical consequence but consider the following circuit in which a secret bit k is being masked by two independent and uniform random variables r_1, r_2 using xor, $k \oplus (r_1 \oplus r_2)$.



Suppose that the attacker is able to physically interfere with the wiring of the circuit in a way that the r_1 variable is sometimes overridden by r_2:



In this scenario, when the fault occurs, the output of the circuit will be $k \oplus (r_2 \oplus r_2) = k$, i.e., the mask will fail.

6.1. Side-channel

Throughout this section, we are focused in studying the following scenario: there is a Boolean formula φ with *plaintext* variables $x \in X$, *secret key* variables $k \in K$, and *random mask* variables $r \in R$, that computes a *ciphertext* $\varphi(X, K, R)$. Typically, the Boolean masks are sampled in a uniform and independent fashion, however in general this might not happen. For this, assume that the masks are sampled according to a probability distribution $\mathbb{P} : \{0,1\}^{|R|} \to [0,1]$, where $|R|$ denotes the cardinality of the set R.

Given a Boolean formula φ with sets of variables X, Y we may denote it by $\varphi(X, Y)$ to reinforce that X, Y are free variables in φ. Furthermore, an instantiation of the variables in X is usually denoted by a bold face letter $\mathbf{X} \in \{0,1\}^{|X|}$. Thus, $\varphi(X, Y)$ where the variables in X are instantiated to $\mathbf{X}$ is denoted by $\varphi(\mathbf{X}, Y)$. We also denote $\{0,1\}^{|X|}$ by $\mathsf{dom}(X)$.

We say that a set S has *polynomial size* in φ, denoted by $|S| = \mathsf{poly}(|\varphi|)$, when there is a positive polynomial p such that $|S| \leq p(|\varphi|)$ and also that $|\psi| \leq p(|\varphi|)$ for all formulas $\psi \in S$.

We denote by $\wp(S)$ the set of subsets of S and by $\wp_{\geq 2}(S)$ the set of subsets of S with cardinality greater than 2.

Definition 4 (Induced Probability Distribution)**.** *Given a Boolean formula $\varphi(X, K, R)$, and $\mathbb{P}$ the probability distribution of R, we denote the* probability distribution induced *by φ with $D_{\mathbb{P},\varphi} : \mathsf{dom}(X) \times \mathsf{dom}(K) \to [0,1]$. Specifically, given a valuation on the plaintexts and keys, $v : X \cup K \to \{0,1\}$, the induced distribution $D_{\mathbb{P},\varphi}(v(X), v(K))$ is a random variable D with probability*

$$\mathsf{P}(D = 1) = \sum_{\mathbf{R} \in \mathsf{dom}(R)} \mathsf{P}(\mathbb{P} = \mathbf{R}) \cdot \bar{v}_{\mathbf{R}}(\varphi) \ ,$$

where $\bar{v}_{\mathbf{R}} : X \cup K \cup R \to \{0,1\}$ extends the valuation v to R as prescribed by $\mathbf{R} \in \mathsf{dom}(R)$.

Example 2. *Consider the Boolean formula* $\varphi \triangleq x \oplus k \oplus (r_1 \oplus r_2)$ *and assume that the random mask variables are uniformly, and independently generated. Then, the induced probability distribution* D *for the valuation* $v(x) = 0, v(k) = 1$ *is*

$$\begin{aligned}\mathsf{P}(D=1) &= \sum_{\mathbf{R}\in\{0,1\}^2} \frac{1}{4}\cdot \bar{v}_{\mathbf{R}}(\varphi) \\ &= \frac{1}{4}\cdot \bar{v}_{(1,1)}(\varphi) + \frac{1}{4}\cdot \bar{v}_{(0,0)}(\varphi) + \frac{1}{4}\cdot \bar{v}_{(1,0)}(\varphi) + \frac{1}{4}\cdot \bar{v}_{(0,1)}(\varphi) \\ &= \frac{1}{4}\cdot 1 + \frac{1}{4}\cdot 1 + \frac{1}{4}\cdot 0 + \frac{1}{4}\cdot 0 \\ &= \frac{1}{2}\ .\end{aligned}$$

Definition 5 (Perfect Masking)**.** *Given a Boolean formula* $\varphi(X, K, R)$ *and* $\mathbb{P}$ *the probability distribution of* R*, we say that* φ *is* perfectly masked *under* $\mathbb{P}$ *if*

$$D_{\mathbb{P},\varphi}(\mathbf{X}, \mathbf{K}) = D_{\mathbb{P},\varphi}(\mathbf{X}, \mathbf{K}')\ ,$$

for any plaintext $\mathbf{X}$ *and secret keys* $\mathbf{K}, \mathbf{K}'$*.*

Consider the following illustrative example of these concepts.

Example 3 ([38])**.** *Consider the following masking functions and their respective output given the values of* k, r_1 *and* r_2 *assuming that* r_1, r_2 *are independent and uniform randomly sampled as defined in Table 1.*

$$\begin{cases} o_1 = k \wedge (r_1 \wedge r_2) \\ o_2 = k \vee (r_1 \wedge r_2) \\ o_3 = k \oplus (r_1 \wedge r_2) \\ o_4 = k \oplus (r_1 \oplus r_2) \end{cases}$$

k	r_1	r_2	o_1	o_2	o_3	o_4
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	0	1	1	0
1	1	0	0	1	1	0
1	1	1	1	1	0	1

Table 1: Boolean masking examples: only o_4 perfectly masks the secret value k.

By inspection of the output probability distribution we observe that it varies depending on k *for all masking functions except for* o_4*. This is the only function in which the output distribution is the same for both* $k = 1$ *and* $k = 0$*. An attacker sampling the function would observe the same ratio of zeros and ones independently of the value* k*, which is not true for the other masking functions.*

Then the problem we want to solve is naturally defined:

Definition 6 (Perfect masking decision problem)**.** *Given a Boolean formula* φ *with plaintexts* X*, secret keys* K *and random mask variables* R *distributed according to* $\mathbb{P}$*, determine if* φ *is perfectly masked under* $\mathbb{P}$*.*

We also remark that this approach to study power side-channel masks implicitly assumes a *Hamming Weight* model for the attacker abilities. This means that an attacker is able to successfully distinguish values with different Hamming weights which is a common model used to study power-related side-channel attacks since they correlate well enough [39]: a computation on variables with a high Hamming weight consumes more power than a computation on variables with a low Hamming weight.

6.2. Modelling perfect masking in GGenPSAT

A formula is perfectly masked if it is not possible to distinguish the distributions on the outputs when different keys $\mathbf{K}, \mathbf{K}'$ are being used. This depends on the Boolean formula φ which encodes the computation or circuit, as well as the joint probability distribution $\mathbb{P}$ of the random masks $r \in R$. In the case that $\mathbb{P}$ can be modelled in a finite set of GGenPSAT formulas, we denote it by $\mathcal{R}(\mathbb{P})$. Then, in the probabilistic formalism of GGenPSAT, testing if a formula is perfectly masked corresponds to determining if the set of formulas,

$$\begin{cases} \mathsf{Pr}(\varphi(X, K, R)) \neq \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1 \text{ for } x \in X \\ \mathsf{Pr}(k) = 0 \vee \mathsf{Pr}(k) = 1 \text{ for } k \in K \cup K' \\ \mathcal{R}(\mathbb{P}) \end{cases} \tag{11}$$

is unsatisfiable, where $\mathcal{R}(\mathbb{P})$ is a set of GGenPSAT formulas that model the probability distribution $\mathbb{P}$ of the random masks. Notice that the case where this formula is unsatisfiable is when there do not exist variables $\mathbf{X}, \mathbf{K}, \mathbf{K}'$ for which the probabilities $\mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K}, R))$ and $\mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K}', R))$ do not coincide and so, the probability distribution is indistinguishable, thus making the computation $\varphi(X, K, R)$ secure i.e., φ is perfectly masked in the sense of [38].

In this sense, it is natural to model the problem of deciding if a computation is perfectly masked as a GGenPSAT problem. Each instance is composed by a set of formulas (11) mainly parametrized by a set of probabilistic formulas $\mathcal{R}(\mathbb{P})$ specifying the probability distribution on the masks to be considered.

As we will make extensive use of the probabilistic formula scheme of the form $\mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1$, which states that x is essentially propositional in nature, we introduce it as an abbreviation

$$\mathsf{prop}(x) \triangleq (\mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1) \ ,$$

which we also extend to sets of variables X in the natural way

$$\mathsf{prop}(X) \triangleq (\bigwedge_{x \in X} \mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1) \ .$$

This way, we restate the GGenPSAT formulation (11) as

$$\mathsf{PM}(\varphi,\mathbb{P}) : \begin{cases} \mathsf{Pr}(\varphi(X,K,R)) \neq \mathsf{Pr}(\varphi(X,K',R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \mathcal{R}(\mathbb{P}) \end{cases} \tag{12}$$

where $\mathcal{R}(\mathbb{P})$ uniquely characterizes the probability distribution $\mathbb{P}$ of the random mask variables in R. We now prove that indeed this formulation corresponds to the problem of deciding if a Boolean formula is perfectly masked under $\mathbb{P}$. For this, we first need an auxiliary lemma.

Lemma 4. *Let $\psi(x,Y)$ be a Boolean formula with free variables $\{x\} \cup Y$. The following implications hold:*

1. $\mathsf{Pr}(x) = 0 \rightarrow \mathsf{Pr}(\psi(x,Y)) = \mathsf{Pr}(\psi(0,Y))$;
2. $\mathsf{Pr}(x) = 1 \rightarrow \mathsf{Pr}(\psi(x,Y)) = \mathsf{Pr}(\psi(1,Y))$.

Proof. To prove case *1.* assume that $\mathsf{Pr}(x) = 0$ and observe that, denoting by π the probability distribution over the valuations on $\{x\} \cup Y$,

$$\begin{array}{rcll} \mathsf{Pr}(\psi(x,Y)) & = & \sum_{v \Vdash \psi(x,Y)} \pi_v & \\ & = & \sum_{v \Vdash \psi(x,Y) \wedge x} \pi_v & + \sum_{v \Vdash \psi(x,Y) \wedge \neg x} \pi_v \\ & = & 0 & + \sum_{v \Vdash \psi(x,Y) \wedge \neg x} \pi_v \\ & = & \sum_{v \Vdash \psi(0,Y)} \pi_v & \\ & = & \mathsf{Pr}(\psi(0,Y))) & . \end{array}$$

The second case follows analogously. □

Proposition 3. *Given a formula $\varphi(X,K,R)$ the* GGenPSAT *problem $\mathsf{PM}(\varphi,\mathbb{P})$ is unsatisfiable iff $\varphi(X,K,R)$ is perfectly masked under $\mathbb{P}$.*

Proof. Assume that the $\mathsf{PM}(\varphi,\mathbb{P})$ is satisfiable and let π be the probability distribution over the set of valuations $\mathcal{V}$ on X, K, K', R that satisfies the constraints in $\mathsf{PM}(\varphi,\mathbb{P})$. Notice that since π satisfies $\mathsf{prop}(X)$ and $\mathsf{prop}(K \cup K')$, we apply Lemma 4 and conclude that there is an assignment $\mathbf{X}, \mathbf{K}, \mathbf{K'}$ for the variables in X, K, K' such that $\mathsf{Pr}(\varphi(X,K,R)) = \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K},R))$ and $\mathsf{Pr}(\varphi(X,K',R)) = \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K'},R))$. Therefore,

$$\pi \Vdash \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K},R)) \neq \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K'},R)) \ .$$

This, on the other hand, means that

$$\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} \pi_v \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K'},R)} \pi_v \ .$$

Since each valuation is only free for the variables in R, and $\pi \Vdash \mathcal{R}(\mathbb{P})$,

$$\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} \pi_v \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K}',R)} \pi_v$$

$$\Leftrightarrow \sum_{\mathbf{R} \in \mathsf{dom}(R)} \pi_{\bar{v}_{\mathbf{R}}} \cdot \bar{v}_{\mathbf{R}}(\varphi(\mathbf{X},\mathbf{K},R)) \neq \sum_{\mathbf{R} \in \mathsf{dom}(R)} \pi_{\bar{v}_{\mathbf{R}}} \cdot \bar{v}_{\mathbf{R}}(\varphi(\mathbf{X},\mathbf{K}',R))$$

$$\Leftrightarrow \sum_{\mathbf{R} \in \mathsf{dom}(R)} \mathsf{P}(\mathbb{P} = \mathbf{R}) \cdot \bar{v}_{\mathbf{R}}(\varphi(\mathbf{X},\mathbf{K},R)) \neq \sum_{\mathbf{R} \in \mathsf{dom}(R)} \mathsf{P}(\mathbb{P} = \mathbf{R}) \cdot \bar{v}_{\mathbf{R}}(\varphi(\mathbf{X},\mathbf{K}',R))$$

$$\Leftrightarrow \mathsf{P}(D_{\mathbb{P},\varphi}(\mathbf{X},\mathbf{K}) = 1) \neq \mathsf{P}(D_{\mathbb{P},\varphi}(\mathbf{X},\mathbf{K}') = 1)$$

and so φ is not perfectly masked under $\mathbb{P}$ according to Definition 5. For the direct implication, observe that assuming the unsatisfiability of $\mathsf{PM}(\varphi, \mathbb{P})$, using Lemma 4, all possible instantiations would need to satisfy $\mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K},R)) = \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K}',R))$ and so, φ would be perfectly masked under $\mathbb{P}$. □

Provided this formulation for deciding if a formula is perfectly masked under $\mathbb{P}$ in GGenPSAT, the computational complexity of this problem becomes, in a way, parametrized by the size of the set of formulas that defines the probability distribution of the random mask variables. We recall that GGenPSAT is NP-complete [17], and thus, if both the number of formulas in $\mathcal{R}(\mathbb{P})$ and their size are polynomially bounded on the size of the original formula φ, the problem of deciding if a formula is perfectly masked, which corresponds to determining the unsatisfiability of $\mathsf{PM}(\varphi, \mathbb{P})$, lies in co-NP.

Proposition 4. *Let $\varphi(X, K, R)$ be a Boolean formula and $\mathbb{P}$ a probability distribution on R. If $|\mathcal{R}(\mathbb{P})| = \mathsf{poly}(|\varphi|)$, then the problem of deciding if φ is perfectly masked under $\mathbb{P}$ is in* co-NP.

6.3. Characterizing attackers with side-channel capabilities

In this section, we formally characterize four types of attackers with side-channel capabilities and study the computational problem of deciding if a Boolean formula is perfectly masked against each different type of attacker. The scenario comprises a Boolean formula which contains plaintext variables X, secret key variables K and random variables R which are uniformly and mutually independent distributed. The goal of the attacker is to be able to distinguish when different keys are being used. We will model four different types of attackers:

1. Passive attacker is only able to observe the result of the computation nodes.
2. Variable-dependency attacker is able to change the dependence and independence of the random masks being used in the computation node.
3. Fault-injection attacker is able to alter the probabilities of each random mask variable, making them biased or even deterministic.
4. General attacker has the power of a variable-dependency attacker as well as a fault-injection attacker.

6.3.1. Perfect Masking against a Passive Attacker

In the case that the random masks are independent and uniformly distributed, this problem is easily reducible to counting the number of satisfying assignments of φ depending on the value of the secret key. If this value differs, the formula is not perfectly masked. In [38], the perfect masking problem is solved by reduction to the satisfiability of a formula in SMT that states that the number of satisfying assignments of φ is independent of the secret parameters. Specifically, the authors define an equivalent formulation for the perfect masking problem as follows: determine if a formula $\varphi(X, K, R)$ is perfectly masked if for all instances of the variables in X, K, $\mathbf{X} \in \mathsf{dom}(X)$ and $\mathbf{K}, \mathbf{K}' \in \mathsf{dom}(K)$

$$\#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}, R)) = \#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}', R)) \ ,$$

where for fixed values of $\mathbf{X}, \mathbf{K}$, $\#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}, R))$ is the number of assignments over $r \in R$ in which the Boolean formula is satisfiable. If this formula is unsatisfiable, it means that there is a plaintext $\mathbf{X}$ and two keys $\mathbf{K}, \mathbf{K}'$ such that the probability distribution of the Boolean formula $\varphi(\mathbf{X}, \mathbf{K}, R)$ differs from $\varphi(\mathbf{X}, \mathbf{K}', R)$. This means that information about the secret key is being leaked and thus the computation is not perfectly masked. We will refer to this as the *perfect masking by counting* problem. Implementations wise, [38] solves $\#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}', R))$ by encoding it in an exponentially sized SMT formula on the number of variables in R.

We now describe how to model a passive attacker in the $\mathsf{GGenPSAT}$ framework. In this case, the random masks are uniform and independent random variables, and so, the set $\mathcal{R}(\mathbb{P})$ is composed of $\mathsf{Pr}(r) = \frac{1}{2}$ for all $r \in R$. Furthermore, we need to specify that all these random variables are mutually independent, i.e., $\mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}}$ for any $S \in \wp_{\geq 2}(R)$.

Example 4. *For a specific example, consider* $\varphi(\{\}, \{k\}, \{r_1, r_2\}) = k \oplus (r_1 \oplus r_2)$. *Thus, the problem of deciding whether this computation is perfectly masked, assuming the random masks are mutually independent and uniform, rests on deciding the satisfiability of the* $\mathsf{GGenPSAT}$ *problem*

$$\begin{cases} \mathsf{Pr}(k \oplus (r_1 \oplus r_2)) \neq \mathsf{Pr}(k' \oplus (r_1 \oplus r_2)) \\ \mathsf{prop}(k) \wedge \mathsf{prop}(k') \\ \mathsf{Pr}(r_i) = \frac{1}{2} \qquad \textit{for } i = 1, 2 \\ \mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{4} \end{cases}$$

which is unsatisfiable as we have seen in Subsection 5.1.2. This means that indeed, this formula is secure under uniform and independent random masks.

In the general case, deciding if a Boolean formula $\varphi(X, K, R)$ is perfectly masked against a passive attacker rests on determining the satisfiability of the following $\mathsf{GGenPSAT}$ problem $\mathsf{PM}(\varphi, \mathsf{Passive})$:

$$\mathsf{PM}(\varphi, \mathsf{Passive}) : \begin{cases} \mathsf{Pr}(\varphi(X,K,R) \neq \mathsf{Pr}(\varphi(X,K',R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \mathsf{Pr}(r) = \frac{1}{2} & \text{for } r \in R \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} & \text{for } S \in \wp_{\geq 2}(R) \end{cases}$$

In the next proposition, we show that this formulation of perfect masking is equivalent to the one introduced by Wang et al. [38] – the perfect masking by counting problem. To do this, we first need to show that

$$\begin{cases} \mathsf{Pr}(r) = \frac{1}{2} & \text{for } r \in R \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} & \text{for } S \in \wp_{\geq 2}(R) \end{cases}$$

defines the joint distribution of $|R|$ independent $\mathsf{Bernoulli}(\frac{1}{2})$ random variables.

Lemma 5. *If a probability distribution π over valuations on a set R satisfies*

$$\begin{cases} \mathsf{Pr}(r) = \frac{1}{2} & \textit{for } r \in R \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} & \textit{for } S \in \wp_{\geq 2}(R) \end{cases}$$

then π is the joint probability distribution of $|R|$ independent $\mathsf{Bernoulli}(\frac{1}{2})$.

Proof. This is a restatement from a result in [40] by noting that these correspond to the cross-moments of the joint random variables and that these $2^{|R|}$ parameters characterize precisely this probability distribution. An alternative formulation of this would be

$$\mathsf{Pr}(\bigwedge_{r \in S} r \wedge \bigwedge_{r \in R \setminus S} \neg r) = \frac{1}{2^{|R|}} \qquad \text{for } S \in \wp(R) \ ,$$

which explicitly defines the full joint probability distribution and therefore fully characterizes the distribution. □

Proposition 5. *Given a formula $\varphi(X,K,R)$ the* GGenPSAT *problem* $\mathsf{PM}(\varphi, \mathsf{Passive})$ *is unsatisfiable iff $\varphi(X,K,R)$ is perfectly masked according to the perfect masking by counting problem.*

Proof. Assume that the $\mathsf{PM}(\varphi, \mathsf{Passive})$ is satisfiable and let π be the probability distribution over the set of valuations $\mathcal{V}$ on X, K, K', R that satisfies the constraints in $\mathsf{PM}(\varphi, \mathsf{Passive})$. Notice that since π satisfies $\mathsf{prop}(X)$ and $\mathsf{prop}(K \cup K')$, we apply Lemma 4 and conclude that there is an assignment $\mathbf{X}, \mathbf{K}, \mathbf{K'}$ for the variables in X, K, K' such that $\mathsf{Pr}(\varphi(X,K,R)) = \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K},R))$ and $\mathsf{Pr}(\varphi(X,K',R)) = \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K'},R))$. Therefore,

$$\pi \Vdash \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K},R)) \neq \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K'},R)) \ .$$

This, on the other hand, means that

$$\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} \pi_v \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K'},R)} \pi_v \ .$$

These only depend on the variables in R and by Lemma 5 we know that π must be uniform on each valuation. Thus,

$$\begin{aligned}
&\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} \frac{1}{2^{|R|}} \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K}',R)} \frac{1}{2^{|R|}} \\
\Leftrightarrow &\frac{1}{2^{|R|}} \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} 1 \neq \frac{1}{2^{|R|}} \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K}',R)} 1 \\
\Leftrightarrow &\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} 1 \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K}',R)} 1 \\
\Leftrightarrow &\#\mathsf{SAT}(\varphi(\mathbf{X},\mathbf{K},R)) \neq \#\mathsf{SAT}(\varphi(\mathbf{X},\mathbf{K}',R)) \ ,
\end{aligned}$$

which shows that φ is not perfectly masked by counting.

For the direct implication, observe that assuming the unsatisfiability of $\mathsf{PM}(\varphi, \mathsf{Passive})$, using Lemma 4, all possible instantiations would need to satisfy $\mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K},R)) = \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K}',R))$ and so, Lemma 5 would imply, $\#\mathsf{SAT}(\varphi(\mathbf{X},\mathbf{K},R)) = \#\mathsf{SAT}(\varphi(\mathbf{X},\mathbf{K}',R))$. □

The number of probabilistic formulas necessary to specify the probability distribution of the random masks is as many as the number of subsets of R, i.e., $2^{|R|}$. This amounts to an exponential sized GGenPSAT instance which is not an improvement regarding the encoding in SMT of [38]. However, as we will see in the next sections, the expressiveness of GGenPSAT allows us to model situations in which the attackers actively interfere with the system. Such situations are intrinsically probabilistic and cannot be easily modelled in SMT or #SAT.

6.3.2. Perfect Masking against a Variable-dependency Attacker

In this section, we study the problem of deciding the perfect masking of a Boolean formula against an attacker which is capable of manipulating the dependency of the different random variables used as masks. These attackers are able to make variables depend on each other, where originally they were independent. In this way, we can construct a family of problems which consist in deciding perfect masking against each different variable-dependency attacker.

In the GGenPSAT formalism, this is done by considering the problem unconstrained by the independency requirements. Consider the following GGenPSAT problem parametrized by the Boolean formula φ as well as the set $\mathcal{A}_{vd} \subseteq \wp_{\geq 2}(R)$, which characterizes the power of the attacker by defining the sets of variables for which the attacker is able to interfere with

$$\mathsf{PM}(\varphi, \mathcal{A}_{vd}) : \begin{cases} \mathsf{Pr}(\varphi(X,K,R)) \neq \mathsf{Pr}(\varphi(X,K',R)) & \\ \mathsf{prop}(X) & \\ \mathsf{prop}(K \cup K') & \\ \mathsf{Pr}(r) = \frac{1}{2} & \text{for } r \in R \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} & \text{for all } S \in \wp_{\geq 2}(R) \setminus \mathcal{A}_{vd} \end{cases}$$

In this setting, an attacker is able to manipulate any variable dependency specified by the set $\mathcal{A}_{vd}$. Also note that a passive attacker is characterized by $\mathcal{A}_{vd} = \emptyset$.

Definition 7. *A Boolean formula is perfectly masked against a variable-dependency attacker $\mathcal{A}_{vd}$ if the* GGenPSAT *problem* $\mathsf{PM}(\varphi, \mathcal{A}_{vd})$ *is unsatisfiable.*

By not imposing any independence restriction, $\mathcal{A}_{vd} = \wp_{\geq 2}(R)$, we model the strongest possible variable-dependency attacker. As noted previously in Proposition 4, with $\mathcal{A}_{vd} = \wp_{\geq 2}(R)$, we obtain $\mathcal{R}(\mathbb{P}) = \{\mathsf{Pr}(r) = \frac{1}{2} \mid r \in R\}$, which has linear size on the size of R, and so this problem becomes co-NP.

Proposition 6. *Given a Boolean formula φ and a variable-dependency attacker $\mathcal{A}_{vd} = \wp_{\geq 2}(R)$, deciding if φ is perfectly masked against $\mathcal{A}_{vd}$ is in* co-NP.

Proof. In this case, determining if φ is perfectly masked against a variable-dependency attacker corresponds to the unsatisfiability of the GGenPSAT problem $\mathsf{PM}(\varphi, \mathcal{A}_{vd})$:

$$\begin{cases} \mathsf{Pr}(\varphi(X, K, R)) \neq \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \mathsf{Pr}(r) = \frac{1}{2} \qquad\qquad \text{for } r \in R \end{cases}$$

which has linear size in the size of φ. This shows the problem is in co-NP since the GGenPSAT problem is in NP. □

Proposition 7. *Consider a Boolean formula φ and a variable-dependency attacker $\mathcal{A}_{vd}$ such that $|\mathcal{P}_{\geq 2}(R) \setminus \mathcal{A}_{vd}| = \mathsf{poly}(|\varphi|)$. The problem of deciding if φ is perfectly masked against $\mathcal{A}_{vd}$ is in* co-NP.

Proof. In this case, we have a GGenPSAT problem with $|\mathcal{P}_{\geq 2}(R) \setminus \mathcal{A}_{vd}| + |R| + 2|K| + |X| + 1$ probabilistic formulas. By hypothesis, this set has polynomial size on $|\varphi|$ and so this problem is in co-NP. □

6.3.3. Perfect Masking against a Fault-injection Attacker

An attacker capable of fault-injection is characterized by its ability to control program variables and program flow by inserting hardware flaws. In our setting, we can model this type of behaviour by describing an attacker that is able to manipulate the random variables of the system. This manipulation can either mean that the attacker can skew the uniform distribution of a random mask r, and, for instance, impose that $\mathsf{Pr}(r) \geq b$, or even to deterministically control a variable, i.e., impose that $\mathsf{Pr}(r) = 0 \vee \mathsf{Pr}(r) = 1$ or even that $\mathsf{Pr}(r) = 1$.

We characterize each fault-injection attacker $\mathcal{A}_{fi}$ by the set of random variables it manipulates, also called the insecure random variables and denoted by $\mathsf{InsR} \subseteq R$. Furthermore, this set is divided in two disjoint sets $\mathsf{InsR} = \mathsf{FIns} \cup \mathsf{PIns}$: the set of fully controlled variables FIns and the set of partially controlled variables, PIns. By a fully controlled variable r, we mean the attacker can impose any probabilistic assertion on the system regarding that variable. Since the modelling is done by specifying what the attacker must adhere to, no GGenPSAT formula is imposed on the FIns variables.

On the other hand, an attacker can only influence a partially controlled variable in a specific manner determined by a set of GGenPSAT formulas involving only variables in PIns, $\Delta(\mathsf{PIns}) \subseteq \mathsf{Prob}(\mathsf{PIns})$ that explicitly defines the control that the attacker has over these variables. For instance an attacker might be able to skew the probability distribution of r, but not know exactly how it changed: $\mathsf{Pr}(r) < 0.2 \vee \mathsf{Pr}(r) > 0.8$. In summary, $\mathcal{A}_{fi} = \langle \mathsf{InsR} = \mathsf{FIns} \cup \mathsf{PIns},\ \Delta(\mathsf{PIns}) \rangle$.

This way, consider the GGenPSAT problem parametrized by a Boolean formula φ as well as an attacker $\mathcal{A}_{fi} = \langle \mathsf{InsR} = \mathsf{FIns} \cup \mathsf{PIns},\ \Delta(\mathsf{PIns}) \rangle$,

$$\mathsf{PM}(\varphi, \mathcal{A}_{fi}) : \begin{cases} \mathsf{Pr}(\varphi(X, K, R)) \neq \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \psi \qquad \text{where } \psi \in \Delta(\mathsf{PIns}) \\ \mathsf{Pr}(r) = \frac{1}{2} \qquad \text{for } r \in R \setminus \mathsf{InsR} \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} \qquad \text{for } S \in \wp_{\geq 2}(R \setminus \mathsf{InsR}) \end{cases}$$

If this problem is unsatisfiable, this means that it is not possible to distinguish the behaviour of the Boolean formula φ when different secret keys are being used, even when manipulating some of the masking variables. In other words, this means that the formula is perfectly-masked against an attacker with fault-injection capabilities.

Definition 8. *A formula φ is perfectly masked against a fault-injection attacker $\mathcal{A}_{fi}$ if the* GGenPSAT *problem $\mathsf{PM}(\varphi, \mathcal{A}_{fi})$ is unsatisfiable.*

6.3.4. Perfect Masking against a general attacker

In this section, we consider a general attacker which can both inject faults, as well change the dependency of the random variables. An attacker is characterized by their ability to

- manipulate probabilities of the random masks in the set InsR. From this set, the attacker can either have full control of the variable or only partial control. This distinction is specified by partitioning the set of insecure masks in FIns and PIns. Furthermore, there is a set of GGenPSAT formulas involving only variables in PIns, $\Delta(\mathsf{PIns}) \subseteq \mathsf{Prob}(\mathsf{PIns})$, that explicitly defines the control that the attacker has over these variables.
- manipulate variable dependency. We denote by $\mathcal{A}_{vd} \subseteq \wp_{\geq 2}(R)$ the subsets of variables for which the attacker is able to manipulate.

Thus, given a Boolean formula φ and an attacker

$$\mathcal{A} = \langle \mathsf{InsR} = \mathsf{FIns} \cup \mathsf{PIns},\ \Delta(\mathsf{PIns}),\ \mathcal{A}_{vd} \rangle,$$

denote by $\mathsf{PM}(\varphi, \mathcal{A})$ the GGenPSAT problem defined by the following formulas

$$\mathsf{PM}(\varphi, \mathcal{A}) : \begin{cases} \mathsf{Pr}(\varphi(X, K, R)) \neq \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \psi \qquad \text{where } \psi \in \Delta(\mathsf{PIns}) \\ \mathsf{Pr}(r) = \frac{1}{2} \qquad\qquad \text{for } r \in R \setminus \mathsf{InsR} \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} \qquad \text{for all } S \in \wp_{\geq 2}(R) \setminus \mathcal{A}_{vd} \end{cases}$$

Definition 9. *A Boolean formula φ is perfectly masked against a general attacker $\mathcal{A}$ if the* GGenPSAT *problem* $\mathsf{PM}(\varphi, \mathcal{A})$ *is unsatisfiable.*

7. Conclusions and Future Work

Throughout this work, we explored a generalized version of probabilistic satisfiability, GenPSAT. Capitalizing on its NP-completeness, we presented a polynomial reduction from GenPSAT to MIP, which was proved to be correct. Since the translated MIP problem only suffers a quadratic growth, we were able to solve reasonably sized instances for different values of the parameters: number of variables, clauses, and probabilistic formulas. Up to this point, our main contribution was on the extension of the language from PSAT to GenPSAT, rather than to optimize the performance of the solver. The lack of sharpness observed in GenPSAT phase transition curves could have been caused by a number of reasons that range from the low number of instances samples for each data point (100) to the translation to MIP that was used. This translation although being correct, might not be the most efficient for this problem. This pertinent issue is handled in our extension GGenPSAT presented in [17] and addressed in Section 5. The GGenPSAT problem naturally models Boolean combinations of linear inequalities involving probabilities of propositional formulas. The phase transition curves for GGenPSAT are presented and discussed in [17].

We concluded the paper with applications to formal-verification, namely satisfiability problems, in the area of cryptography, specifically on side-channel analysis and their mitigation techniques. A common, and usually efficient technique that mitigates power-related side-channel attacks is Boolean masking: these techniques work by applying a random (and unknown) mask to a computation step, in order to *mask* the secret keys and other values that are used in said computation. However, if these masks are not designed in a proper way, even when they are correctly applied, they could leak information regarding the secrets they are designed to hide. In this work, we modelled the problem of deciding if a Boolean formula is perfectly masked in the probabilistic formalism. Furthermore, we generalized this scenario to encompass an active attacker interfering with the Boolean masks, by changing dependency between random variables, or actually changing their probability distributions. Remarkably, we found that solving the problem of deciding if a Boolean formula is perfectly masked is (in terms of computational complexity) easier to solve in the presence of more powerful attackers, lowering the complexity of this problem to co-NP.

In practical terms, this shows that it is easier to decide if a formula is perfectly masked when powerful attackers are considered and we are actually gaining more security guarantees by solving an easier problem. However, naturally, the set of formulas which are secure against active attackers is much smaller than when a passive adversary is considered.

References

[1] C. Caleiro, F. Casal, A. Mordido, Generalized Probabilistic Satisfiability, Electronic Notes in Theoretical Computer Science 332, LSFA 2016.

[2] S. A. Cook, The complexity of theorem-proving procedures, in: Proceedings of 3rd ACM symposium on Theory of computing, ACM, 1971, pp. 151–158.

[3] A. Biere, M. Heule, H. van Maaren, Handbook of satisfiability, Vol. 185, IOS press, 2009.

[4] L. De Moura, N. Bjørner, Satisfiability modulo theories: introduction and applications, Communications of the ACM 54 (9) (2011) 69–77.

[5] G. Boole, Investigation of The Laws of Thought On Which Are Founded the Mathematical Theories of Logic and Probabilities, 1853.

[6] N. J. Nilsson, Probabilistic logic, Artif. Intell. 28 (1) (1986) 71–88.

[7] M. Finger, G. Bona, Probabilistic satisfiability: Logic-based algorithms and phase transition., in: IJCAI, IJCAI/AAAI, 2011, pp. 528–533.

[8] G. Georgakopoulos, D. Kavvadias, C. H. Papadimitriou, Probabilistic satisfiability, Journal of Complexity 4 (1) (1988) 1 – 11.

[9] F. G. Cozman, L. F. Ianni, Probabilistic satisfiability and coherence checking through integer programming (2013) 145–156.

[10] G. Bona, F. G. Cozman, M. Finger, Generalized probabilistic satisfiability through integer programming, Journal of the Brazilian Computer Society 21 (1) (2015) 1–14.

[11] M. Finger, G. Bona, Probabilistic satisfiability: algorithms with the presence and absence of a phase transition, Annals of Mathematics and Artificial Intelligence 75 (3) (2015) 351–389.

[12] A. Mordido, C. Caleiro, Probabilistic logic over equations and domain restrictions, accepted in Mathematical Structures in Computer Science.

[13] R. Fagin, J. Y. Halpern, N. Megiddo, A logic for reasoning about probabilities, Inf. Comput. 87 (1-2) (1990) 78–128.

[14] C. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover Publications, 1982.

[15] V. Chandru, J. Hooker, Optimization methods for logical inference, John Wiley and sons, Inc, 1999.

[16] P. Cheeseman, B. Kanefsky, W. M. Taylor, Where the really hard problems are., in: IJCAI, Vol. 91, 1991, pp. 331–340.

[17] C. Caleiro, F. Casal, A. Mordido, Classical generalized probabilistic satisfiability, in: Proceedings of IJCAI-17, 2017, pp. 908–914.

[18] J. Rosenhouse, The Monty Hall problem: the remarkable story of Math's most contentious brain teaser, Oxford University Press, 2009.

[19] F. Casal, A. Mordido, C. Caleiro, GenPSAT solver, available online at https://github.com/fcasal/genpsat.git (2016).
[20] V. Chvátal, Linear programming, Macmillan, 1983.
[21] G. S. Tseitin, On the complexity of derivations in the propositional calculus, Studies in Mathematics and Mathematical Logic Part II (1968) 115–125.
[22] I. P. Gent, T. Walsh, The hardest random SAT problems, Springer, 1994.
[23] I. Gurobi Optimization, Gurobi optimizer reference manual (2015).
[24] F. Casal, A. Mordido, C. Caleiro, GGenPSAT solver, available online at https://github.com/fcasal/ggenpsat.git (2016).
[25] P. Baltazar, Probabilization of Logic Systems, Ph.D. thesis, IST - Technical University of Lisbon, Portugal (2010).
[26] A. Mordido, A probabilistic logic over equations and domain restrictions, Ph.D. thesis, IST - Universidade de Lisboa, Portugal (2017).
[27] P. C. Kocher, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems, Springer Berlin Heidelberg, 1996, pp. 104–113.
[28] P. C. Kocher, J. Jaffe, B. Jun, Differential power analysis, in: Proceedings of CRYPTO '99, Springer-Verlag, 1999, pp. 388–397.
[29] K. Gandolfi, C. Mourtel, F. Olivier, Electromagnetic Analysis: Concrete Results, Springer Berlin Heidelberg, 2001, pp. 251–261.
[30] D. Agrawal, B. Archambeault, J. R. Rao, P. Rohatgi, The EM Side—Channel(s), Springer Berlin Heidelberg, 2003, pp. 29–45.
[31] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, ECDH key-extraction via low-bandwidth electromagnetic attacks on pcs, in: Cryptographers' Track at the RSA Conference, Springer, 2016, pp. 219–235.
[32] C. Ramsay, Tempest attacks against AES (White paper from Fox-IT, 2017).
[33] S. Nauerth, M. Fürst, T. Schmitt-Manderbach, H. Weier, H. Weinfurter, Information leakage via side channels in freespace BB84 quantum cryptography, New Journal of Physics 11 (6) (2009) 065001.
[34] D. Boneh, R. DeMillo, R. Lipton, On the importance of checking cryptographic protocols for faults, in: EUROCRYPT'97, Springer, 1997.
[35] P. Dusart, G. Letourneux, O. Vivolo, Differential fault analysis on AES, in: Applied Cryptography and Network Security, Springer, 2003, pp. 293–306.
[36] S. Chari, C. S. Jutla, J. R. Rao, P. Rohatgi, Towards Sound Approaches to Counteract Power-Analysis Attacks, Springer, 1999, pp. 398–412.
[37] E. Prouff, M. Rivain, Masking against Side-Channel Attacks: A Formal Security Proof, Springer Berlin Heidelberg, 2013, pp. 142–159.
[38] H. Eldib, C. Wang, P. Schaumont, Formal verification of software countermeasures against side-channel attacks, ACM Transactions on Software Engineering and Methodology (TOSEM) 24 (2) (2014) 11.
[39] S. Mangard, E. Oswald, T. Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security), Springer-Verlag New York, Inc., 2007.
[40] J. L. Teugels, Some representations of the multivariate bernoulli and binomial distributions, Journal of Multivariate Analysis 32 (2) (1990) 256 – 268.

Appendix A.

This paper extends [1]. As such, Sections 2-4 are included in [1]. Then, Section 5, partially integrates [17] up until Subsection 5.1. All the rest of the paper, from Subsection 5.1 onwards, is new material.