
SATMC: a SAT-based Model Checker for Security Protocols

Luca Compagna

joint work with Alessandro Armando



AI-Lab - DIST - University of Genova

JELIA, Lisbon, 27-30 Sept 2004



Automated Validation of Internet Security Protocols and Applications

Shared cost RTD (FET open) project IST-2001-39252

Motivations

- The world is **distributed**.

E-commerce, wireless communication, ...

1. $A \rightarrow B : \{A, N_A\}_{K_B}$
2. $B \rightarrow A : \{N_A, N_B\}_{K_A}$
3. $A \rightarrow B : \{N_B\}_{K_B}$

- **Security is critical**: breaches can be ruinous!

- Protocols are **cornerstone** of security.

Key distribution, authentication, ...

- Unfortunately, even simple protocols are **difficult** to get right by simple inspection. Therefore, protocol designers need help!

- **Idea**: use **Automated Reasoning techniques** for analyzing security protocols.

Why SAT?

- **Context:** Dramatic speed-up of SAT solvers in the last decade: problems with thousands of variables are now solved routinely in milliseconds.

This has led to breakthroughs in planning and hardware verification.

Why SAT?

- **Context:** Dramatic speed-up of SAT solvers in the last decade: problems with thousands of variables are now solved routinely in milliseconds.

This has led to breakthroughs in planning and hardware verification.

- **Approach:** we have investigated if similar results can be obtained by applying SAT-based model-checking to security protocols.

Protocol Analysis: Modeling

- Protocol as a **state transition system** in which states correspond to information possessed by participating agents.
- **Intuitively:**
 - **honest agents** that communicate over an channel in order to achieve **security requirements** (e.g. exchange some secret information), but
 - the **channel is insecure** and controlled by an **intruder** (overhear, intercept, and send fraudulent messages).

Protocol Analysis: Security Problems

- Specified by means of the IF rule-based language
 - **state**: *set of facts*
 - **transition relation**: *labeled rewrite rules*. E.g.,

$$ik(\{M\}_K).ik(K^{-1}) \xrightarrow{\text{decrypt}(K,M)} ik(M).ik(\{M\}_K).ik(K^{-1})$$

Protocol Analysis: Security Problems

- Specified by means of the IF rule-based language
 - **state**: *set of facts*
 - **transition relation**: *labeled rewrite rules*. E.g.,

$$ik(\{M\}_K).ik(K^{-1}) \xrightarrow{\text{decrypt}(K,M)} ik(M).ik(\{M\}_K).ik(K^{-1})$$

- Security requirements such as **authentication** and **secrecy** are reduced to **reachability problems** on this model.
- The general verification problem is **undecidable**: we focus on **reachability problem with finite number of sessions**.
- This is adequate in practice as **attacks** on well-known protocols often exploit a **small number of sessions**.

Protocol Security Problem as SAT problems

Bounded model-checking of security protocols via reduction to SAT with iterative deepening on the number of steps.

Theorem: Given, a Protocol Security Problems Π and a positive integer k , we build a SAT formula Φ_{Π}^k such that any **model of Φ_{Π}^k** corresponds to **attacks on Π** .

$$\Phi_{\Pi}^k = I_0 \wedge \bigwedge_{k=0}^{k-1} T_i^{i+1} \wedge G_k$$

The formula Φ_{Π}^k represents the search space up to depth equal to k .

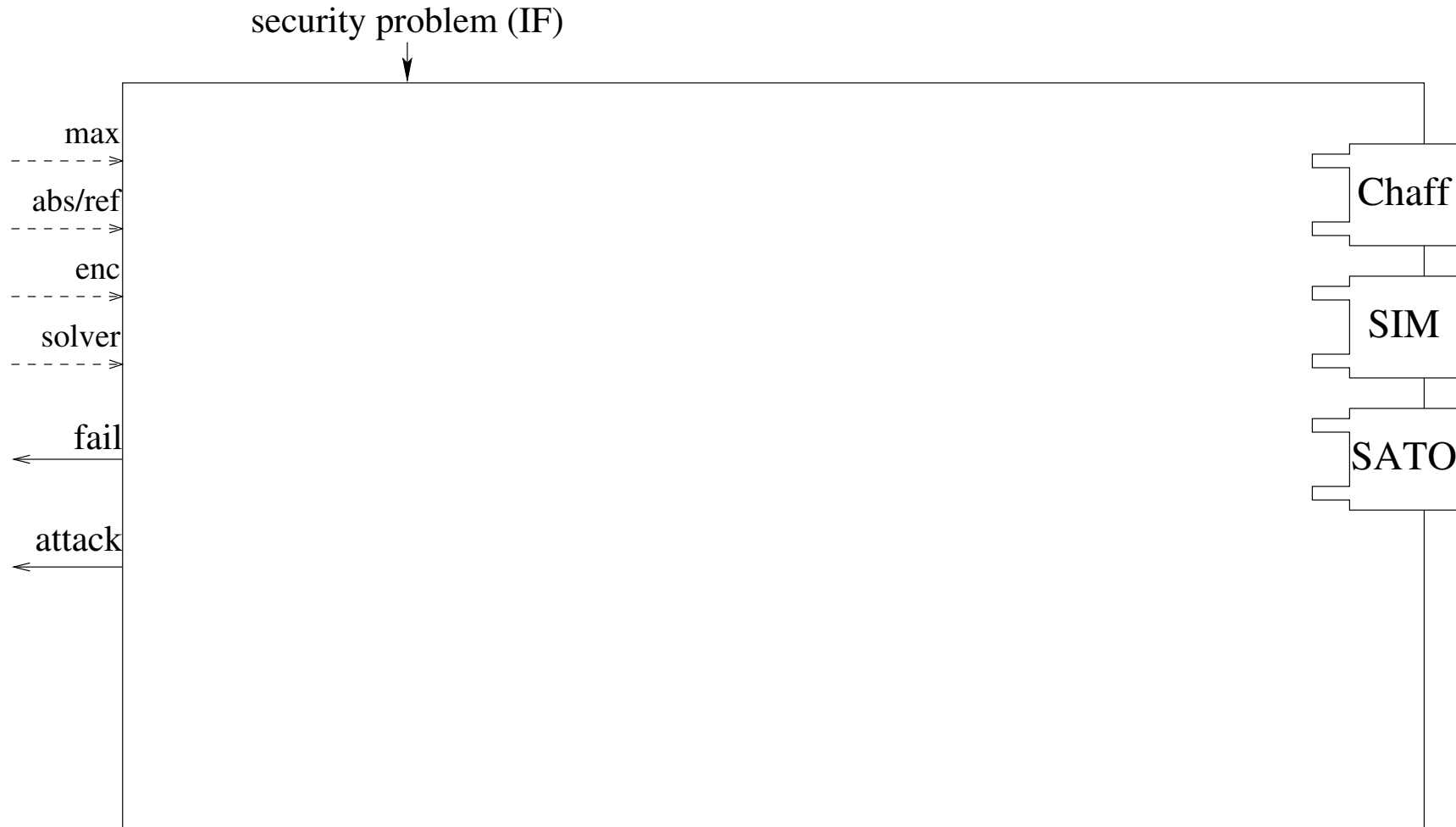
Implementation

SATMC v2.0:

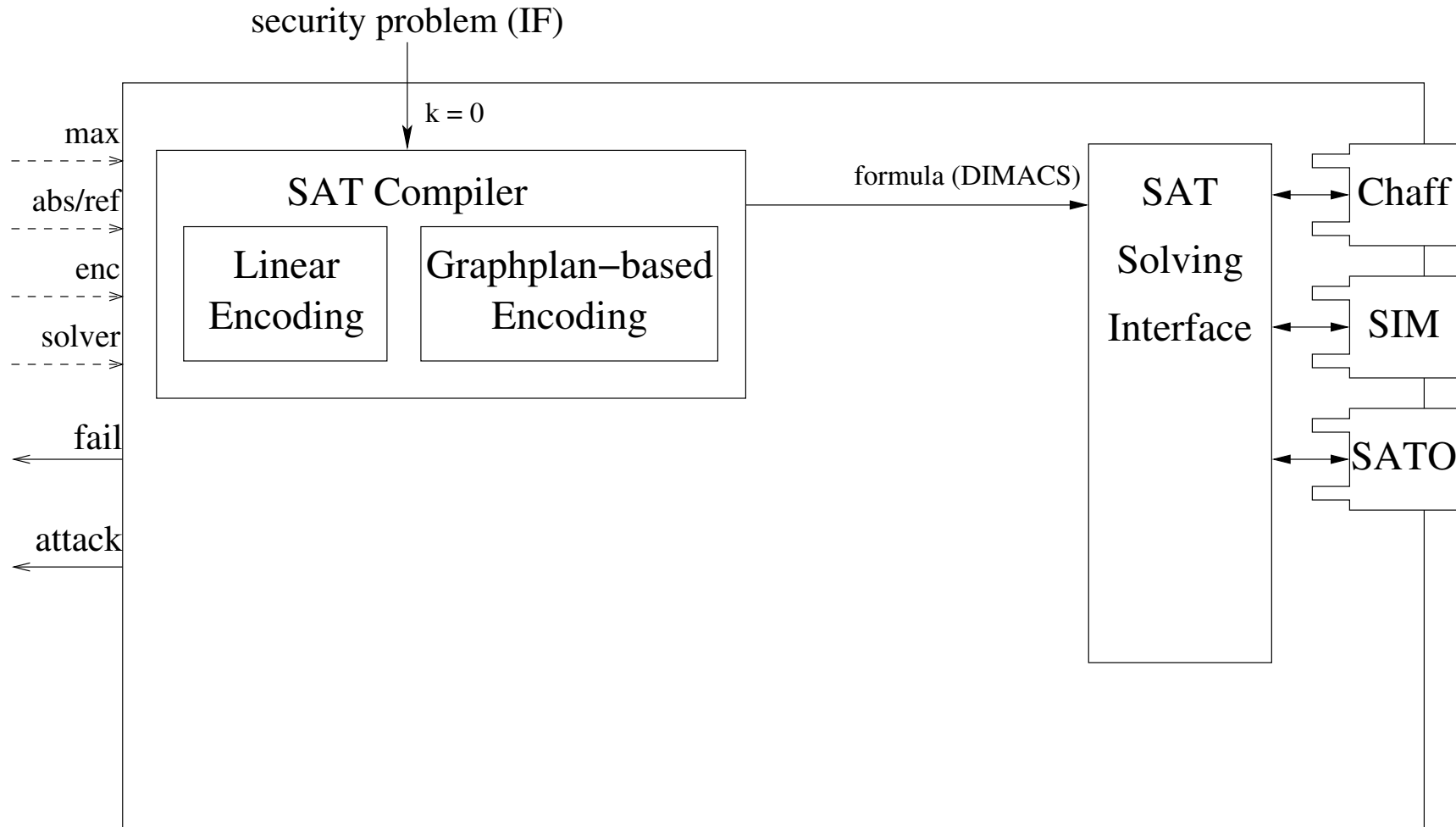
- input specification in **IF** rule-based language;
- set of **optimizing transformations** to get encodings of manageable size;
- **linear** and **graphplan-based encodings** with **iterative deepening** on the number of steps;
- **abstraction/refinement strategy** based on neglecting mutex relations;
- a translator from **IF** to **LPARSE** specifications which are fed into the logic program solvers **CMODELS** and **SMODELS**. (see talk on [Wednesday](#).)

Download it at: <http://www.ai.dist.unige.it/satmc>

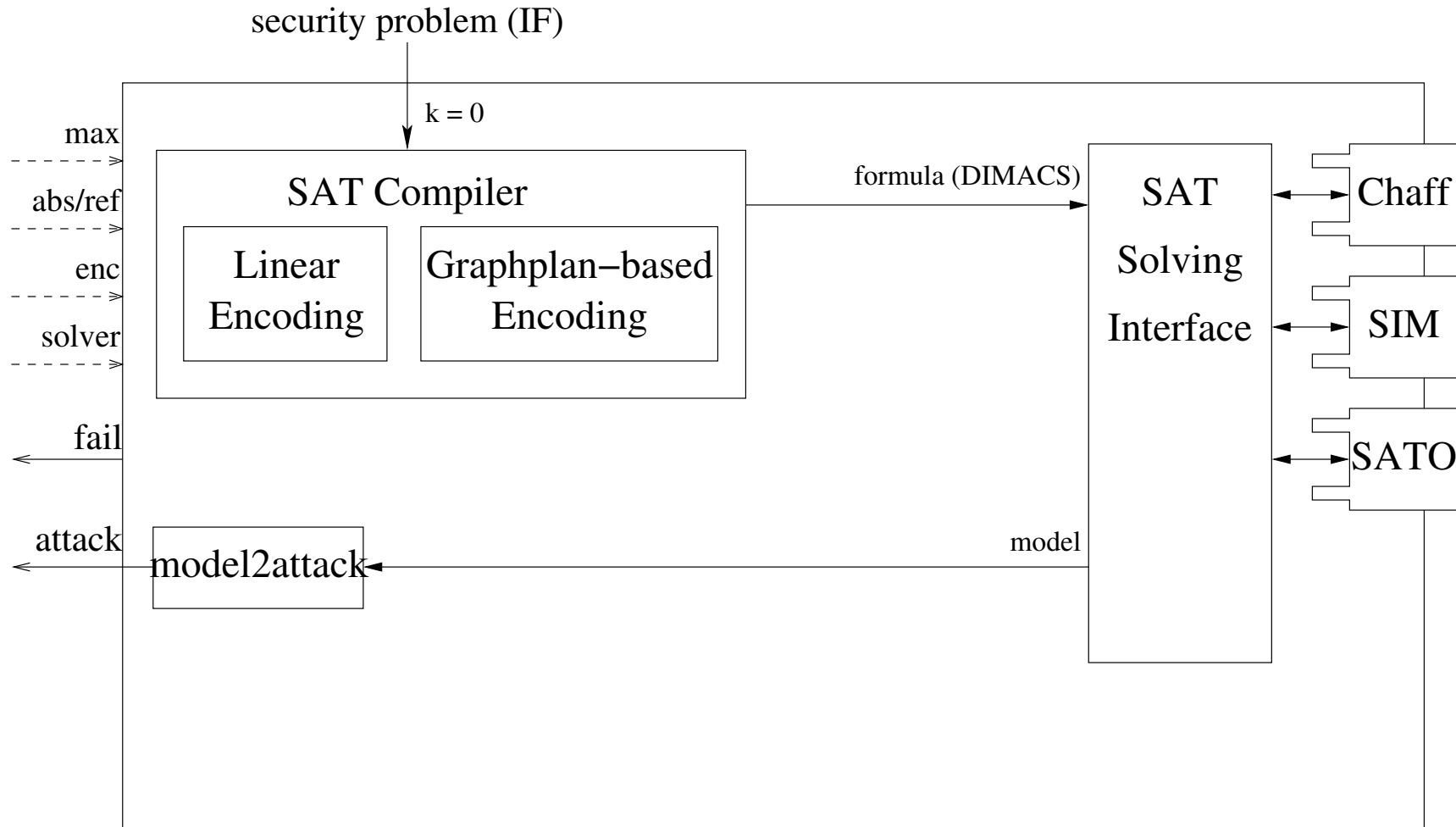
Architecture



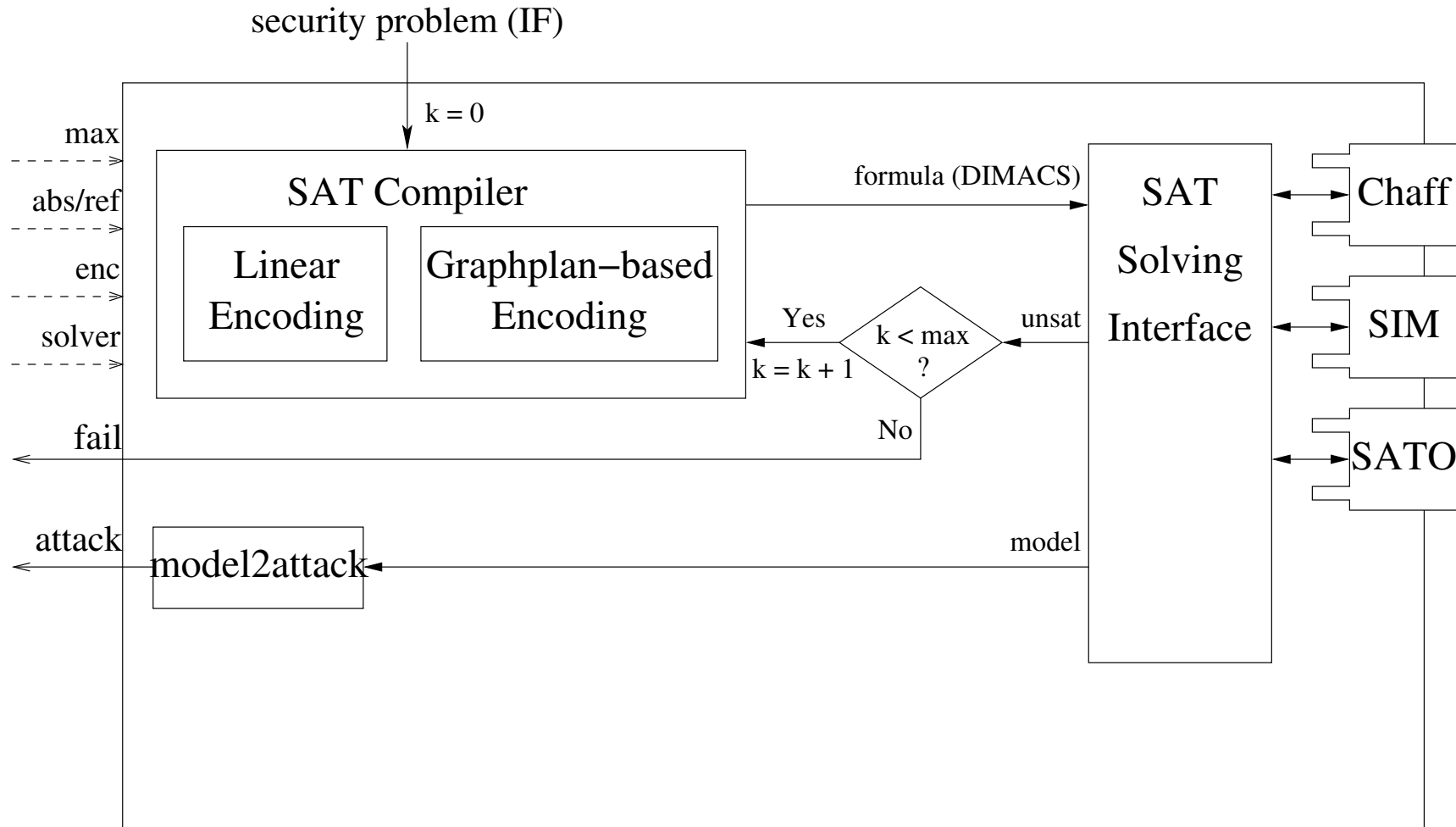
Architecture



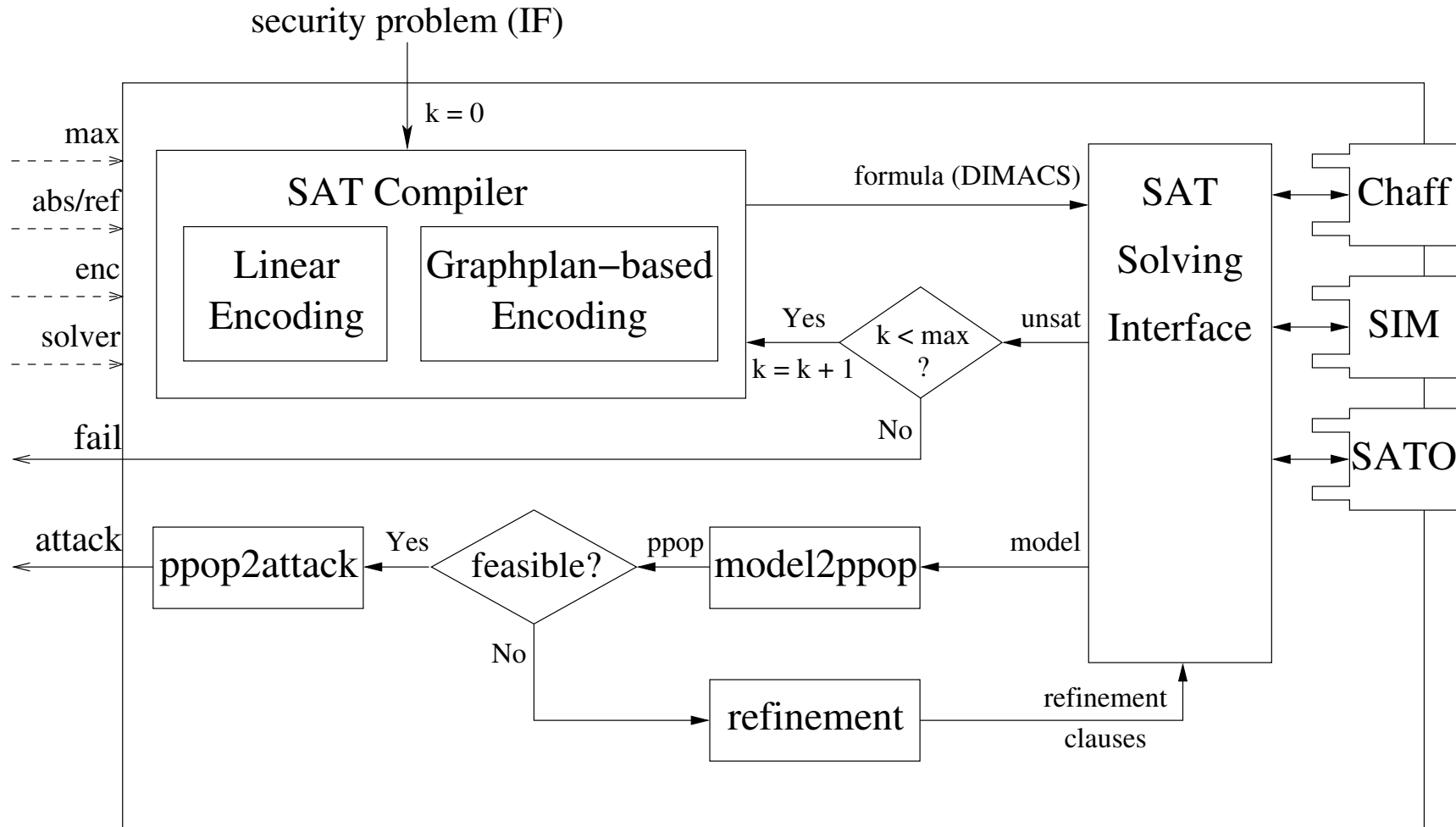
Architecture



Architecture



Architecture



Experimental Results* on Clark/Jacob library

Protocol	K	A	CL	EncT	SolT
<i>Andrew</i>	9	442	1,365	0.14	0.01
<i>EKE</i>	5	394	1,337	0.12	0.00
<i>ISO-CCF-1 U</i>	4	102	295	0.00	0.00
<i>ISO-CCF-2 M</i>	4	115	311	0.02	0.00
<i>ISO-PK-1 U</i>	4	149	418	0.03	0.00
<i>ISO-PK-2 M</i>	4	129	363	0.02	0.00
<i>ISO-SK-1 U</i>	4	93	265	0.01	0.00
<i>ISO-SK-2 M</i>	4	117	314	0.02	0.00
<i>KaoChow 1</i>	7	426	1,781	0.18	0.01
<i>KaoChow 2</i>	9	726	3,393	0.32	0.00
<i>KaoChow 3</i>	9	990	6,118	0.66	0.03

Protocol	K	A	CL	EncT	SolT
<i>KLS rep.</i>	7	1,634	23,190	3.89	0.03
<i>NSCK</i>	9	435	1,406	0.12	0.00
<i>NSPK</i>	7	411	1,279	0.09	0.00
<i>NSPK-server</i>	8	847	2,702	0.23	0.00
<i>SPLICE</i>	9	951	3,168	0.32	0.00
<i>Swick 1</i>	5	192	554	0.06	0.00
<i>Swick 2</i>	6	257	838	0.08	0.00
<i>Swick 3</i>	4	171	498	0.05	0.01
<i>Swick 4</i>	5	215	634	0.04	0.00
<i>Stubblebine rep</i>	3	146	478	0.04	0.00
<i>Woo-Lam M</i>	6	481	1,539	0.19	0.00

(*) Experiments have been carried out on a PC with a 1.4 GHz CPU, 1 GB of RAM, and with max=10, abs/ref=false, enc=graphplan and solver=chaff.

Conclusions and Perspectives

- Proposed an **open and flexible platform** for **automatically compiling security protocols into SAT**.
- Assessed the **effectiveness** of SATMC on the Clark/Jacob library.
- Building Encryption Properties (e.g. $g^{x*y} = g^{y*x}$) using **axioms** (preliminary results).
- Abstraction techniques for **unbounded verification** (work in progress).
- **AVISPA** project: **industrial-strength** technology for the **A**utomated **V**erification of **l**arge-**s**cale **I**nternet **S**ecurity **P**rotocols and **A**pplications.

Thanks for you attention