Binary decision diagrams for security protocols

dr hab. inż. Henryk Piech, dr Mirosław Kurkowski, dr inż. Olga Siedlecka-Lamch

Instytut Informatyki Teoretycznej i Stosowanej
Politechnika Częstochowska

4 czerwca 2012 roku

dr hab. inż. Henryk Piech, dr Mirosław Kurkowski, dr inż. Olga Siedlecka-Lamch

-

BDD definition

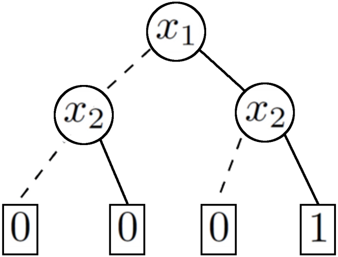
Definition

An BDD G representing the Boolean Functions $f_1, \ldots, f_m$ over the variables $x_1, \ldots, x_n$ is a directed acyclic graph with following properties:

1. Nodes without outgoing edges, which are called sinks or terminal nodes, are labeled by 0 or 1.
2. All non-sink nodes of G, which are also called internal nodes, are labeled by a variable, a nd have two outgoing edges, a 0-edge and 1-edge.
3. On each directed path in the OBDD each variable occurs at most once as the label of the node.

Simple example

x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1



BDD types

- OBDD
- OBDD with complemented edges
- Algebraic Decision Diagrams
- Zero-suppressed Binary Decision Diagrams

OBDD definition

Definition

An OBDD G representing the Boolean Functions $f_1, \ldots, f_m$ over the variables $x_1, \ldots, x_n$ is a directed acyclic graph with following properties has all properties of BDD and

1. there is a variable ordering π - a permutation of $x_1, \ldots, x_n$ and on each directed path the variables occur according to this ordering

Basic operations I

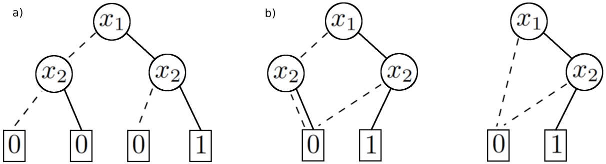
1. **Evaluation**: For an OBDD G representing f and an input a compute the value $f(a)$.
2. **Reduction**: For an OBDD G compute the equivalent reduced OBDD.
3. **Equivalence test**: Test whether two functions represented by OBDDs are equal.
4. **Satisfiability problems**: These problems include:
   - Satisfiability: For an OBDD G representing f find an input a for which $f(a) = 1$ or output that no such input exists.
   - SAT-Count: For an OBDD G representing f compute the number of inputs a for which $f(a) = 1$.
5. **Synthesis** (also called Apply): For functions f and g represented by an OBDD G include into G a representation for $f \otimes g$ where $\otimes$ is a binary Boolean operation (e.g., $\wedge$).

Basic operations II

6. **Replacements** (also called Substitution): There are two replacement operations:
   - Replacement by constants: For a function f represented by an OBDD, for a variable x_i and a constant $c \in 0, 1$ compute an OBDD for $f_{|x_i=c}$.
   - Replacement by functions: For functions f and g represented by an OBDD and for a variable x_i compute an OBDD for $f_{|x_i=g}$.
7. **Universal quantification and existential quantification**: F or a function f represented by an OBDD and for a variable x_i compute an OBDD for $(\forall x_i : f) := f_{|x_i=0} \wedge f_{|x_i=1}$ or $(\exists x_i : f) := f_{|x_i=0} \vee f_{|x_i=1}$, respectively.

Reduction

x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1



Knowledge variables

Needham Schroeder Public Key Protocol:

$$\begin{array}{llll} \alpha_1 & A & \rightarrow B & : \langle N_A \cdot i(A) \rangle_{K_B}, \\ \alpha_2 & B & \rightarrow A & : \langle N_A \cdot N_B \rangle_{K_A}, \\ \alpha_3 & A & \rightarrow B & : \langle N_B \rangle_{K_B}. \end{array} \tag{1}$$

knowledge variables:

$$\begin{array}{lll} x_A^{N_A} & - & (N_A \in Know_A), \quad x_A^{N_B} - (N_B \in Know_A), \\ x_B^{N_A} & - & (N_A \in Know_B), \quad x_B^{N_B} - (N_B \in Know_B). \end{array} \tag{2}$$

If α_i^j is i-th step in the j-th execution of the protocol, then the variable which corresponds to this step is marked by $x_{\alpha_i^j}$.

Boolean functions

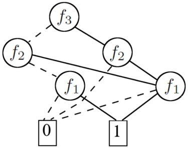
$$
\begin{aligned}
f_1^1 &= x_A^{N_A} \wedge x_B^{N_A} \wedge x_{\alpha_1^1}, \\
f_2^1 &= x_B^{N_B} \wedge x_B^{N_A} \wedge x_A^{N_B} \wedge x_{\alpha_2^1}, \\
f_3^1 &= x_A^{N_B} \wedge x_{\alpha_3^1}.
\end{aligned} \tag{3}
$$

OBDD construction

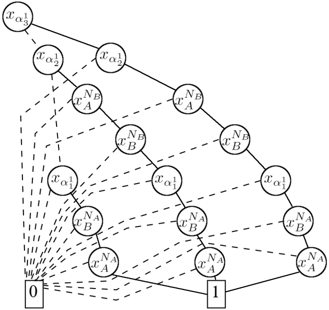
a)

f_3	f_2	f_1	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

b)



c)



Boolean functions for Low's atack

BDD for SP

dr hab. inż. Henryk Piech, dr Mirosław Kurkowski, dr inż. Olga Siedlecka-Lamch

Introduction

Security protocols

Needham-Schroeder protocol

Low's atack

Formal model

Binary decision diagrams

Definition and types of BDD

Algorithms

OBDD for security protocols

Boolean functions

OBDD construction

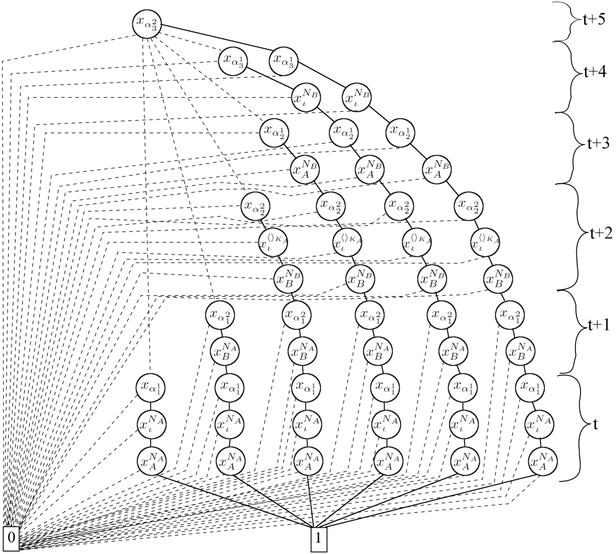
Low's atack:

$$
\begin{array}{llll}
\alpha_1^1 & A & \rightarrow \iota & : \langle N_A \cdot \iota(A) \rangle_{K_\iota}, \\
 & \alpha_1^2 & \iota(A) \rightarrow B & : \langle N_A \cdot \iota(A) \rangle_{K_B}, \\
 & \alpha_2^2 & B \rightarrow \iota(A) & : \langle N_A \cdot N_B \rangle_{K_A}, \\
\alpha_2^1 & \iota & \rightarrow A & : \langle N_A \cdot N_B \rangle_{K_A}, \\
\alpha_3^1 & A & \rightarrow \iota & : \langle N_B \rangle_{K_\iota}, \\
 & \alpha_3^2 & \iota(A) \rightarrow B & : \langle N_B \rangle_{K_B}.
\end{array}
\tag{4}
$$

bollean functions:

$$
\begin{aligned}
f_1^1 &= x_A^{N_A}(t) \wedge x_\iota^{N_A}(t) \wedge x_{\alpha_1^1}(t), \\
f_1^2 &= x_B^{N_A}(t+1) \wedge x_{\alpha_1^2}, \\
f_2^2 &= x_B^{N_B}(t+2) \wedge x_\iota^{\langle N_A \cdot N_B \rangle_{K_A}}(t+2) \wedge x_{\alpha_2^2}(t+2), \\
f_1^2 &= x_A^{N_B}(t+3) \wedge x_{\alpha_2^2}(t+3), \\
f_1^3 &= x_\iota^{N_B}(t+4) \wedge x_{\alpha_3^2}(t+4), \\
f_2^3 &= x_{\alpha_3^3}(t+5).
\end{aligned}
\tag{5}
$$

OBDD construction



Chains

Definition

The chain in the OBDD tree for the run $\mathfrak{r}$ is called the reduced correct sequence of boolean functions: $\mathfrak{c} = f_{k_1}^{i_1}, f_{k_2}^{i_2}, f_{k_3}^{i_3}, \ldots, f_{k_s}^{i_s}$.

The chain $\mathfrak{c} = f_{k_1}^{i_1}, f_{k_2}^{i_2}, f_{k_3}^{i_3}, \ldots, f_{k_s}^{i_s}$ can be written as:
$\mathfrak{c} = f_{k_1}^{i_1}(t_1) < f_{k_2}^{i_2}(t_2) < f_{k_3}^{i_3}(t_3) < \ldots < f_{k_s}^{i_s}(t_s)$ where $t_m < t_n$, for $m = 1, \ldots, s-1$ and $n = 2, \ldots, s$.

Threat template for Low's atack

$$\begin{aligned} St \quad = \quad & (e_1 = x_A^{N_A}(t)) < t_1 = th < (e_2 = x_{\alpha_1^1}(t)) < t_2 = Th < \\ & (e_3 = x_B^{N_A}(t' > t)) < t_3 = th < (e_4 = x_B^{N_B}(t' > t)) < \\ & t_4 = th < (e_5 = x_{\alpha_2^2}(t' > t)) < t_5 = Th < \\ & (e_6 = x_A^{N_B}(t'' > t')) < t_6 = th < (e_7 = x_{\alpha_1^3}(t'' > t')) \end{aligned} \tag{6}$$

Threat template searching

$$
\begin{aligned}
RS \quad = \quad & (r_1 = x_A^{N_A}(t)) < tr_1 = th < (r_2 = x_{\alpha_1^1}(t)) < tr_2 = th < \\
& (r_3 = x_\iota^{N_A}(t^1 > t)) < tr_3 = th < (r_4 = x_{\alpha_1^2}(t^1 > t)) < \\
& tr_4 = th < (r_5 = x_B^{N_A}(t^2 > t)) < tr_5 = th < \\
& (r_6 = x_B^{N_B}(t^2 > t^1)) < tr_6 = th < (r_7 = x_{\alpha_2^2}(t^2 > t^1)) < \\
& tr_7 = th < (r_8 = x_\iota^{\langle N_A \cdot N_B \rangle_{K_A}}(t^3 > t^2)) < tr_8 = th < \\
& (r_9 = x_{\alpha_2^1}(t^3 > t^2)) < tr_9 = th < (r_{10} = x_A^{N_B}(t^4 > t^3)) < \\
& tr_{10} = th < (r_{11} = x_{\alpha_1^3}(t^5 > t^4)) < tr_{11} = th < \\
& (e_{12} = x_\iota^{N_B}(t^6 > t^5)) < t_{12} = th < (e_{13} = x_{\alpha_3^2}(t^6 > t^5))
\end{aligned}
$$

References I

- Akers, S.B.: Binary decision diagrams. IEEE Trans Comp 27, 509-516 (1978)
- Bryant, R.E.: Binary decision diagrams and beyond: enabling techniques for formal verification. Int Conf CAD, 236-243 (1995)
- Drechsler, R., Becker, B.: Binary decision diagrams - theory and implementation. Kluwer Academic Publishers, Boston, Mass., USA (1998)
- Kurkowski, M., Srebrny, M.: A Quantifier-free First-order Knowledge Logic of Authentication, Fundamenta Informaticae, vol. 72, pp. 263-282, IOS Press 2006
- Kurkowski, M., Penczek, W.: Verifying Security Protocols Modeled by Networks of Automata, Fundamenta Informaticae, Vol. 79 (3-4), pp. 453-471, IOS Press 2007

References II

- Kurkowski, M., Penczek, W.: Verifying Timed Security Protocols via Translation to Timed Automata, Fundamenta Informaticae, vol. 93 (1-3), pp. 245-259, IOS Press 2009