RAPPORT TECHNIQUE EVA

L'outil de vérification HERMES

Auteur	:	Liana Bozga, Yassine Lakhnech, Michaël Périn Verimag
Date	:	mai 2002
Rapport EVA numéro	:	6
Version	:	1

TRUSTED LOGIC S.A.
5 rue du Bailliage
78000 Versailles, France
www.trusted-logic.fr

Laboratoire Spécification Vérification
CNRS UMR 8643, ENS Cachan
61, avenue du président-Wilson
94235 Cachan Cedex, France
www.lsv.ens-cachan.fr

Laboratoire Verimag
CNRS UMR 5104,
Univ. Joseph Fourier, INPG
2 av. de Vignate,
38610 Gières, France
www-verimag.imag.fr

Adresse : VERIMAG,
Centre équation,
2 av. de Vignate,
38610 Gières, France

L'outil de vérification HERMES

Liana Bozga, Yassine Lakhnech, Michaël Périn
Verimag

mai 2002

1 HERMES, un outil pour vérifier les protocoles d'échanges de secrets

HERMES est un outil dédié à la vérification de propriétés de secret pour les protocoles cryptographiques. Étant donné un protocole et une propriété de secret spécifiés dans le langage EVA, HERMES calcule des conditions suffisantes pour garantir la propriété de secret.

2 Architecture et caractéristiques de l'outil

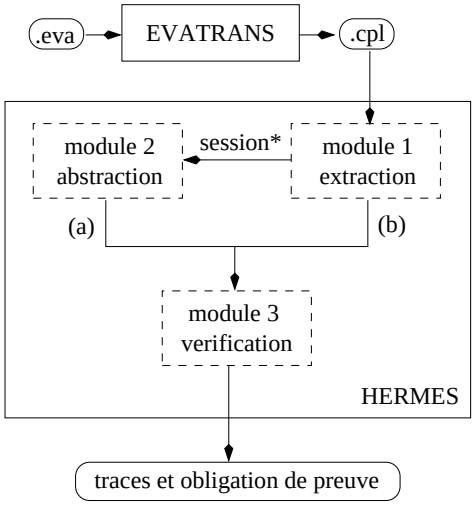

FIG. 1 – *L'architecture d'*HERMES *et la connection au traducteur* EVATRANS

HERMES prend en entrée un fichier `.cpl` produit par le traducteur EVATRANS à partir d'un fichier `.eva`. De cette description il extrait la propriété de secret et la spécification du protocole (module 1 fig. 1). Il calcule une abstraction de la propriété et du protocole (module 2) puis complètent l'ensemble des secrets et génèrent des contraintes qui définissent les conditions dans lesquelles le protocole peut être utilisé sans risque d'invalider la propriété de secret (module 3). L'outil produit également des

traces qui justifient l'ajout de chaque nouveau secret et de chaque nouvelle contrainte. Ces traces correspondent à des tentatives d'attaques.

Le principe de vérification implanté dans HERMES est décrit en détail dans [2], nous en donnons ici une présentation succinte ; nécessaire à l'interprétation des résultats.

Les contraintes calculées par le module 3 d'HERMES s'interprètent comme des limites imposées sur les connaissances de l'Intrus au moment de débuter une session du protocole. La section 8.2 présente un exemple de contraintes. Ces contraintes sont établies de manière à garantir que les messages échangés par des participants honnêtes au cours d'une session du protocole jouée en parallèle avec un nombre arbitraire d'autres sessions, ne permettent pas à l'Intrus de découvrir les secrets échangés par les participants honnêtes.

La méthode de calcul des contraintes du module 3 s'applique aussi bien à un *nombre arbitraire de sessions parallèles* (entrée a) qu'à un *nombre fini de sessions parallèles* (entrée b). Dans le premier cas, le module 2 d'HERMES construit une abstraction qui satisfait la propriété suivante : si la propriété abtraite est vérifiée pour le protocole abtrait alors la propriété de secret est satisfaite par le protocole concret pour un nombre arbitaire de sessions exécutées en parallèles par un nombre arbitraire de participants générant un nombre arbitraire de données fraîches (nonces et clefs).

3 Principe de l'outil

HERMES calcule des conditions suffisantes pour garantir que les messages échangés au cours d'une session du protocole ne peuvent être exploités pour déduire des secrets. L'outil est basé sur la notion de messages capables de garder un secret, appelés messages protecteurs. Il s'agit des messages de la forme $\{X\}_K$, c'est-à-dire cryptés au moyen d'une clef dont l'inverse n'est pas connu de l'Intrus. Les messages protecteurs se déduisent des clefs déclarées secrètes dans les hyptohèses du protocole.

Étant donné un protocole, un ensemble S de secrets et un ensemble d'hypothèses sur les clefs secrètes, HERMES calcule, d'une part, un ensemble de messages qui protègent les secrets échangés au cours du protocole ; et d'autre part, il ajoute aux secrets les messages qui permettent de construire une attaque. Il s'agit en particulier des messages dans lesquels les secrets ne sont pas protégés. À l'issue du calcul dont la terminaison est assurée par un opérateur de convergence forcée (widening), HERMES retourne un ensemble de secrets S' qui contient les secrets de départ et un ensemble de messages protecteurs H'. Les résultats H' et S' définissent les conditions dans lesquelles le protocole peut être utilisée sans risque : la propriété de secret est garantie si, dans les messages initialement connus de l'Intrus, les secrets S' sont protégés par les messages H'.

4 Implémentation actuelle et extensions à venir

Dans sa version actuelle, HERMES traite uniquement les clefs typées. Il n'est donc pas capable de détecter les attaques de type sur les protocoles d'Otway-Rees et de Neumann-Stubblebine. Cependant, la méthode présentée dans [2] peut traiter le cas des *clefs composées* telles qu'elles sont définies dans la sémantique du langage EVA. Cette caractéristique permet de levée la restriction et de détecter les attaques de type mais aussi les attaques reposant sur la composition de clefs à partir de la connaissance

des termes qui permettent d'engender la clef. L'extension de la méthode aux clefs composées n'est pas implémentée dans toute sa généralité, néanmoins, dans le cas du protocole d'Otway-Rees, HERMES permet de détecter l'attaque classique de type.

5 Intégration d'HERMES au projet EVA

Les contraintes retournées par HERMES définissent les conditions d'utilisation du protocole. Dans le cadre du projet EVA, nous adoptons le point de vue inverse puisque les conditions initiales sont fixées. Elles définissent les connaissances (ou non connaissances) initiales de l'Intrus, les contraintes d'HERMES doivent alors être comprise comme une obligation de preuve.

Dans le langage EVA, les conditions initiales sur les connaissances de l'Intrus sont exprimées par des déclarations de la forme `assume *A*G secret(K)` qui indique que la clef *K* est supposée ne jamais être connue de l'Intrus. Nous exploitons ces hypothèses pour satisfaire l'obligation de preuve retournée par HERMES. Étant donnée la forme particulière des hyptohèses, la preuve requise par HERMES est soit triviale et la propriété de secret est satisfaite, ou bien elle est impossible, le plus souvent de manière évidente, et il s'agit d'une attaque. Ce dernier cas se traduit généralement par le fait que l'ensemble de secrets complété par HERMES contient des messages qui sont trivialement constructibles par l'Intrus : ils contiennent uniquement des données publiques telles que des noms de participants, des clés publiques et aucun élément frais. HERMES a été conçu pour détecter au plus tôt ce type d'attaques. Lorsqu'il doit ajouter aux secrets un message qui est trivialement constructible par l'Intrus, il signale une attaque et produit la trace des opérations nécessaires à l'attaque (dans le modèle abstrait).

Du fait des approximations effectuées par l'opérateur de widening, nous ne pouvons éviter le phénomène des fausses attaques. Il s'agit des cas où HERMES produit une trace d'une attaque abstraite qui ne correspond pas à une attaque exécutable dans le modèle réel.

6 Expérimentations

Le tableau 6 présente les résultats obtenus pour la vérification de propriétés de secret sur les protocoles de la base de tests EVA [1]. La mention *« correct »* correspond au cas où l'obligation de preuve généré par HERMES est démontrable à partir des hyptohèses de la spécification. La mention *« attaque »* correspond à une attaque réelle détectée par HERMES. Il existe un troisième cas possible, dit *« inconclusif »*. Il englobe les phénomènes de fausses attaques et d'obligations de preuves non démontrables, qui résultent tout deux des approximations nécessaires à la vérification. Ce cas ne s'est pas présenté sous les hypothèses des protocoles de la base de test EVA.

7 Contribution à la sémantique du langage EVA

Le modèle d'exécution des protocoles sur lequel s'appuie HERMES est conforme à la sémantique du langage EVA [3]. L'extension de notre méthode de vérification au cas des clefs composées a ammené à préciser la sémantique du langage EVA.

1. L'attaque de type ne peut être détectée par la version actuelle d'HERMES qui traite uniquement de clefs typées. Cette restriction sera levée dans la prochaine version qui prendra en compte les clefs composées.

Protocole	Résultat	Temps (s)	Nombre de transitions du protocole abstrait
Yahalom	correct	13,81	35
Needham-Schroeder	attaque	0,04	9
Needham-Schroeder-Lowe	correct	0,03	9
Otway-Rees	correct[1]	0,03	35
Kao-Chow_1	correct	0,13	50
Kao-Chow_2	correct	16,89	370
Neumann-Stubblebine	correct[1]	0,04	75
TMN	attaque	2,78	290
TMN-Lowe	attaque	3,04	325

TAB. 1 – *Résultats des expérimentations menées sur la base de test EVA*

8 Un exemple de vérification effectuée avec HERMES

Le protocole d'Otway-Rees décrit ci-après illustre la forme actuelle des spécifications dont la traduction est acceptée en entrée d'HERMES.

8.1 Spécification d'entrée en langage EVA du protocole

```
Otway_Rees
  A, B, S : principal
  N, Na, Nb : number

  basetype key
  Kas, Kbs, Kab : key

  shr (principal,principal) : key
  alias Kas = shr(A,S)
  alias Kbs = shr(B,S)

  A knows A, B, Kas
  B knows B, S, Kbs
  S knows S, Kas, Kbs

  {
    1. A -> B : N, A, B, {Na, N, A, B}_Kas
    2. B -> S : N, A, B, {Na, N, A, B}_Kas, {Nb, N, A, B}_Kbs
    3. S -> B : N, {Nb, Kab}_Kbs
    4. S -> A : N, {Na, Kab}_Kas
  }

  s. session* {Kas,Kbs} A=A, B=B, S=S

  assume *A*G secret (Kas@s.A),
         *A*G secret (Kbs@s.B),
         *A*G secret (Kas@s.S),
         *A*G secret (Kbs@s.S)
```

```
claim  *A*G secret (Kas@s.A),
       *A*G secret (Kbs@s.B),
       *A*G secret (Kas@s.S),
       *A*G secret (Kbs@s.S),
       *A*G secret (Kab@s.S)
```

8.2 Contraintes retournées par HERMES

```
Secrets : Kas; Kbs; Nb; Kab

Good Patterns:      Bad Patterns:
 {Xs}_Kas;           vide
 {Xs}_Kbs;
 {Xs}_Kab;
```

Interprétation des résultats L'ensembles des secrets S' retournés par HERMES ne contient pas de nouveaux secrets. L'ensemble H' des messages protecteurs retournés par HERMES est représenté comme la différence entre les instances de Good patterns et celles des *Bad Patterns* : les messages protecteurs de secrets sont ceux qui matchent avec les *Good Patterns* privés des messages qui matchent avec les *Bad Patterns*. Ainsi, dans le cas présent, les patterns indiquent que tout message chiffré avec l'une des clefs `Kas`, `Kbs` ou `Kab` peut sans danger contenir un secret.

Obligation de preuve retournée par HERMES Pour valider la propriété de secret, il reste à prouver que sous les hypothèses de la partie «`assume...`», les secrets S' sont protégés par l'ensemble H' de messages retournés par HERMES. Les arguments qui permettent de satisfaire l'obligation de preuve sont les suivants:

- les secrets `Nb` et `Kab` sont générés durant le protocole, ils sont donc frais et par définition, ils ne peuvent apparaître dans aucun message préexistants ; ils sont donc trivialement protégés par n'importe quel message protecteur.
- d'après les hypothèses, les secrets `Kas` et `Kbs` ne sont jamais envoyés (cela n'interdit pas que ces clefs aient été utilisées pour crypter un message). Ces deux secrets sont donc trivialement protégés par n'importe quel message protecteur.

On peut alors conclure que la propriété de secret est vérifiée (dans le cas du protocole avec clefs typées). Notons que les obligations de preuve produites par HERMES se résolvent en général par des raisonnements similaires aux précédents qui font appel à des arguments de fraîcheur ou bien directement aux hypothèses.

Références

[1] Dominique Bolignano, Francesca Fiorenza, Florent Jacquemard, and Daniel Le Métayer. EVA test base. Technical Report EVA-4, Trusted Logic S.A., November 2001. Version 1.17.

[2] L. Bozga, Y. Lakhnech, and M. Périn. Pattern-based abstraction for verifying secrecy in protocols, 2002. Submitted for publication.

[3] J. Goubault-Larrecq. Les syntaxes et sémantique du langage de spécification EVA. Technical Report EVA-12?, LSV, November 2001.