Research Center for Information Security
情報セキュリティ研究センター
Research Center for Information Security

実用的な暗号の形式的証明に向けて:
アセンブリでの多倍長整数アルゴリズムの形式的検証

アフェルト レナルド　　ソフトウェアセキュリティ研究チーム

セキュリティのための最も厳密な検証手法

形式的検証とは？→国際規格(コモンクライテリアなど)による最も厳密なセキュリティレベル:

1. ソフトウェアの実装の正しさの機械検証
2. 第三者による全自動検査可能な証明の発行

暗号とは？→セキュリティプロトコルの基本的な部分:

- 数論による情報処理
- 高い効率を要求

⇒

- ◆ 先進アルゴリズム
- ◆ 低レベル実装

代表的な暗号:RSAやElGamalなど

証明理論に基づくコンピュータ上での検証

定理証明器とは？証明理論に基づく開発環境:

- 型理論に基づくプログラミング
- 一階述語論理などによる自動検証
- 高階述語論理(帰納法など)による推論

} 機械検証

- ◆ 型付き高階ラムダ計算の証明オブジェクト

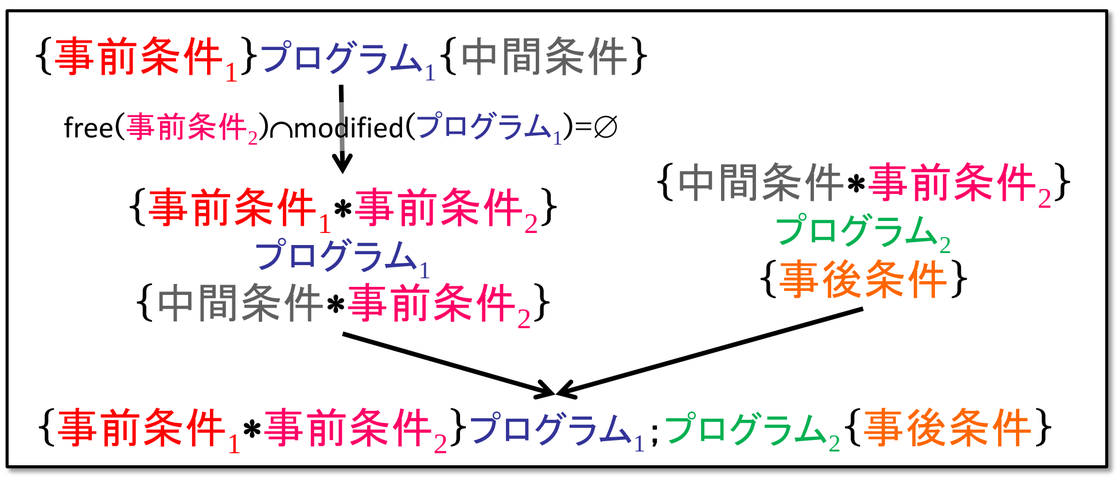
} 第三者による全自動検査可

代表的な定理証明器:Coq (http://www.lix.polytechnique.fr/coq/)

我々の貢献

アセンブリでの整数論アルゴリズムの検証ライブラリ:

- Smartcard向けのアセンブリ (SmartMIPS, MIPS Technologies)
- 低レベルデータ構造の扱い
  - ◆ 文字型やポインターや多倍長整数など
- ホーア論理による証明の構成
  - ◆ その論理自体の健全性の機械検証など
  - ◆ 分離論理による形式的証明の組み合わせ:



- 実用的な暗号関数の実装とその形式的証明 - - - - - - - - - ->
- 形式的検証済みの翻訳など

応用例:
効率的なアセンブリでの冪剰余の形式的検証

プログラムの規模:

アセンブリ関数	命令数*
Montgomery冪剰余	100
冪乗ループ	13
Montgomery掛算	87
Montgomery 掛算(元)	36
多倍長引算	20
その他(多倍長比較, 初期化)	17

*nopでない命令 , 繰り返し呼び出しのインラインなし

形式的証明の組み合わせによる形式的検証:

Montgomery冪剰余

$\{A \equiv \beta^k [M] \wedge A' \equiv \beta^{2k} [M] \wedge X,A,A' < M \wedge \text{odd } M \wedge |e| < 32 \wedge M_0 \times \alpha \equiv -1 [\beta]\}$

```
montgomery k x a' x' m ;
addiu i 1 0_16 ;
while (bgez i) (
  montgomery k a a a' m ;
  srlv ei e i ;
  andi ei ei 1_16 ;
  ifte_beq ei, gpr_zero thendo
    (xor a a a' ;
     xor a' a a' ;
     xor a a a')
  elsedo
    montgomery k a' x' a m ;
  addiu i i -1_16) ;
montgomery k a one' a' m
```

$\{\exists A', A' \equiv X^e [M] \wedge A' < M\}$

Montgomery掛算

$\{X,A' < M \wedge M_0 \times \alpha \equiv -1 [\beta]\}$

```
multizero x';
mflhxu gpr_zero ;
mthi gpr_zero ;
mtlo gpr_zero ;
montgomery_raw k x a' x' m ;
ifte_beq C, gpr_zero thendo
  multi_lt k x' m ;
  ifte_beq ret, gpr_zero thendo
    multisub k x' m x'
elsedo
  addiu t t 4_16 ;
  sw C 0_16 t ;
  addiu Sk k 1_16 ;
  multisub Sk x' m x'
```

$\{\exists X', \beta^k X' \equiv X \times A' [M] \wedge \underline{X' < M}\}$

Montgomery掛算(元)

$\{X,A' < M \wedge M_0 \times \alpha \equiv -1 [\beta]\}$

...

$\{\exists X', \beta^k X' \equiv X \times A' [M] \wedge \underline{X' < 2 \times M}\}$

実験結果の一覧

多倍長関数	仕様(略術)	形式的証明行数	
		合計	事前・事後条件だけ
多倍長足算	$C := A + B, B := A + B$	240	77
多倍長引算	$C := A - B, B := A - B, A := A - B$	472	166
多倍長掛算	$C := A \times B$	570	213
Montgomery掛算(元)	$Z := X \times Y [M], Z := X^2 [M] (Z < 2M)$	1239	468
初期化	$A := 0$	137	25
多倍長比較	$A < B$	422	96
Montgomery掛算	$Z := X \times Y [M], Z := X^2 [M] (Z < M)$	474	183
Montgomery冪剰余	$Z := X^e [M]$	616	233
BBS疑似乱数生成器	$Z := X^2 [M] \& 1 \| X^4 [M] \& 1 \| \ldots$	843	298

形式的検証済みのSmartMIPSでの多倍長関数

参考文献

- R. Affeldt, D. Nowak, and K. Yamada, *Certifying Assembly with Formal Cryptographic Proofs: the Case of BBS*, 投稿中
- R. Affeldt, *A Library for Formal Verification of Low-level Programs*, http://staff.aist.go.jp/reynald.affeldt/coqdev/
- アフェルト　レナルド, 「Coq上での組み込み用途の冪剰余の検証」, PPL2009
- R. Affeldt and N. Marti, *An Approach to Formal Verification of Arithmetic Functions in Assembly*, ASIAN 2006
- J.C. Reynolds, *Separation Logic: A Logic for Shared Mutable Data Structures*, LICS 2002

今後の課題

- 大規模のプログラムの形式的検証に向けて拡張
- 高レベルプログラミング言語との形式的なインターフェイス
- PKCS#1に対するRSAの実装の形式的検証

メール先:reynald.affeldt@aist.go.jp; 平成21年4月28日