User-Guide
for Algebraic Intruder Deductions in OFMC

Sebastian Mödersheim
Information Security Group, Dep. of Computer Science, ETH Zurich, Switzerland
www.infsec.ethz.ch/~moedersheim

February 13, 2006

OFMC is now enhanced to include the support i(e)8(or)1-4user-defined algebraic

Symbol	Arity	Intuition	Intruder-Accessible
inv	1	private-key of given public-key	no
crypt	2	asymmetric encryption	yes
scrypt	2	symmetric encryption	yes
pair	2	pairing/concatenation	yes
apply	2	function application	yes
exp	2	exponentiation modulo fixed prime p	yes

Anexample ofsucha specificationcanbe found inAppendix A. This is also the basis for considering offline-guessing attacks [2].

-

There may be more solutions, if $T1$ or $T2$ are themselves terms with *xor* at

```
 Analysis:
   decana(xor(X1,X2))=
     [X1]->[X2]
     [xor(X1,X3)]->[xor(X2,X3)]
```

The last line adds the case that the intruder knows $xor(X1, X3)$, i.e. he

4 Dealing with the Complexity

A The SRP Protocol

The SRP protocol (Secure Remote Passwords, [3]) is a challenging example for algebraic properties, since it requires a full arithmetic theory to work. It uses modular addition, multiplication and exponentiation, and without the necessary properties it is not executablece In the EU project AVISPA, as part of which OFMC and several other toole have be1(ce)n deceelop1(ce)d, this protocol was modeled in a drastically simplifiece version, basically receucing it to a Di□1(ce)-H1(ce)llman key-exchange.

A.1 A Arithmetic Theory

With the new theory features of OFMC, it ie now possiblec267(to)-267(mo)-28(d)1(e)-1(l)-266(the)-267(p)1(rot b1(ce)tween addition, multiplication, and 1(ce)xponentiationYsti1(n)anc928(t)1hatdu theoryi(le)-341(with)-33(thle)-341njecaoryprloparte s(.)447(Wy)84le cnsi(d)1(e)-1rhthleowgingprop1(ceert)1(ie thhtoneag(e)-1(n)9(tos)-1,nonallnifnthemttscor:y

```
  mult(X,one)=X
  mult(mult(X,Y),minv(Y))=X
Topdec:
  % add is associative and commutative:
  topdec(add,add(T1,T2))=
    [T1,T2]
    [T2,T1]
    if T1==add(Z1,Z2){
      [Z1,add(Z2,T2)]
      [add(Z1,T2),Z2]
      if T2==add(Z3,Z4){
        [add(Z1,Z3),add(Z2,Z4)]}}
    if T2==add(Z1,Z2){
      [add(T1,Z1),Z2]
      [Z1,add(T1,Z2)]}
  %
  % mult is associative and commutative:
  topdec(mult,50X,YT1,T2))=
    [T1,T2]
    [T2,T1]
    if T1==50X,YZ1,Z2){
      [Z1,50X,YZ2,T2)]
      [50X,YZ1,T2),Z2]
      if T2==50X,YZ3,Z4){
        [50X,YZ1,Z3),50X,YZ2,Z4)]}}
    if T2==50X,YZ1,Z2){
      [50X,YT1,Z1),Z2]
      [Z1,50X,YT1,Z2)]}
  %
  % Distributivity: mult(X1,add(X2,X3))=add(mult(X1,X2),50X,YX1,X3))
  topdec(add,mult(X1,X2))=
    if X2==add(X3,X4){
      [50X,YX1,X3),mult(X1,X4)]}
  % The ‘‘other direction’’ we currently cannot model, here is how
  % it shall look like in the future:
  %    topdec(mult,add(X1,X2))=
  %      if X1==50X,YX3,X4){
  %        if X2==50X,YX3,X5){
  %          [X3,50X,YX4,X5)]}}
  %
  % Relation between exp,mult and add:
  % expYexpYX1,X2),X3)=expYX1,50X,YX2,X3))
  % expYX1,sum(X2,X3))=50X,YexpYX1,X2),expYX1,X3))
  topdec(exp,expYT1,T2))=
    [T1,T2]
    if T1==expYZ1,Z2){
     [Z1,mult(T2,Z2)]
     [exp(Z1,T2),Z2]}
    if T2==mult(Z1,Z2){
     [exp(T1,Z1),Z2]}
  topdec(mult,exp(T1,T2))=
    if T2==sum(Z1,Z2){
    [exp(T1,Z2),exp(T1,Z2)]}
Analysis:
  decana(add(X1,X2)) =[X1]->[X2]
  decana(mult(X1,X2))=[X1]->[X2]
  decana(exp(X1,X2)) =[X2]->[X1]
  decana(neg(X))=[]->[X]
  decana(minv(X))=[]->[X]
```

Note that with such a theory, several larger protocols will just explode, so only use this theory when you really want to go deep into arithmetic!

A.2 The Protocol Formalization

An important aspect of the protocol that we currently cannot model is the fact that the shared passwords of Users and Hosts, denoted `passwd(User,Host)`, may be weak (guessable). Though foundational research in this direction has been done, for instance [2], this is not yet implemented: it requires algebraic properties 8(instan)1rs ans ans ans an2433(an)tanhn82o33(an)1(cdn)1ronthis messages that contain g^b anyway, it does not make a differe9ce whether this