Compact RSA inclusion/exclusion proofs

(Prerequisite: RSA Accumulators for Plasma Cash history reduction)

Here is a brief summary of my understandings on how compact RSA inclusion/exclusion proofs work. Correct me if I am wrong somewhere.

Inclusion proof

To prove some prime number α exists in $[g...A]$ you provide a cofactor x which satisfies the following equation:

$g^{\alpha x} \equiv A$ (mod N)

If $A \equiv g^{Q}$, then $x=\frac Q \alpha$

Proof:

$\alpha x=\alpha \frac {Q} \alpha=Q$

(I can't find a cofactor x if α doesn't exist in the accumulator, because in that case Q is not divisible by α)

The problem is that x can get very large. We can do a trick here.

We know that any positive integer, including x, can be represented as follows:

$x = B\lfloor \frac x B \rfloor + x \mod B$

where B is an arbitrary positive integer.

Now let's define:

$h=g^\alpha$

$b=h^{\lfloor \frac x B \rfloor} \mod N$

$r=x \mod B$

Now you can say: $b^B.h^r \equiv g^{ax}$ (mod N)

Because:

$b^B.h^r \equiv h^{\lfloor \frac x B \rfloor.B}.h^r \equiv h^{\lfloor \frac x B \rfloor.B + r} \equiv h^x \equiv g^{ax} \equiv A$

(mod N

)

So here we can use the tuple (b, r)

instead of x

as the proof. The benefit of this approach is that both $b < N$

and $r < B$

are constant-sized.

Exclusion proof

To prove some prime number α

does not

exist in [g...A]

you provide a cofactor x

and remainder s

(Where $0 < s < \alpha$

) which satisfy the following equation:

$g^{\alpha x + s} \equiv A$

(mod N

)

If $A \equiv g^{Q}$

, then $s=(Q \mod \alpha)$

, and $x=\frac {Q-s} \alpha$

Proof:

$\alpha x+s=\alpha \frac {Q-s} \alpha + s=Q-s+s=Q$

(I can’t find a remainder $0<s<\alpha$

if α

does exist

in the accumulator)

Like the inclusion proofs, x

can get very large. We can do the same trick here.

Again we represent cofactor x

as follows:

$x = B\lfloor \frac x B \rfloor + x \mod B$

Now let’s say:

$h=g^\alpha$

$b=h^{\lfloor \frac x B \rfloor} \mod N$

$r=x \mod B$

Now you can say: $b^B.h^r.g^s \equiv g^{ax+s}$

(mod N

)

Because:

$b^B.h^r.g^s \equiv h^{\lfloor \frac x B \rfloor.B}.h^r.g^s \equiv h^{\lfloor \frac x B \rfloor.B + r}.g^s \equiv h^x.g^s \equiv g^{ax}.g^s \equiv g^{ax+s} \equiv A$

(mod N

)

So here we can use the tuple (b, r, s)

instead of (x, s)

as the proof. The benefit of this approach is that both $b < N$

and $r < B$

and $s < \alpha$

are constant-sized.

How to choose B?

It seems that the prover is able to create invalid inclusion/exclusion proofs if B

is not set large enough.

As an example, let's say we have 3 prime numbers $\{3,5,11\}$

in our accumulator.

Therefore: $A = g^{3511} \mod N$

Let's say I want to prove that the accumulator includes prime 5

.

In the old approach I would provide $x=33$

as the proof and the user could check the validity of my proof by checking if $g^{5*33} \equiv A$

. I can't prove the accumulator has prime 7

in it as I can't find a x

such that $7x=165$

.

No way for me to cheat the user!

Now suppose I am using the new approach and I want to cheat the user and say the accumulator has number 7 in it. I should give him the tuple (b, r)

such that $b^B.h^r \equiv A$

. I set $B=79$

on purpose.

Here we have $B=79$

and $h=g^7$

Therefore: $b^B.h^r \equiv b^{79}.g^{7r}$

I pick $b = g^{2}$

and $r=1$

So: $(g^{2})^{79}.g^{7*1} \equiv g^{165} \equiv A$

I successfully proved that 7 exists in A

while it is not!

We can force the prover to use a large, deterministic value for B

depending on g

and A

to make it extremely hard for the prover to find an invalid (b, r)

proof. Let’s use a hash function here:

B = hash(g, A)

As @gakonst mentioned, original Wesolowski’s paper states that B

should be a prime number. Don’t know why yet.