Toward the correctness of TweetNaCl's Ed25519 with VST

Benoît Viguier MSc

(λ x y. x@y.nl) benoit viguier

https://www.viguier.nl

DeepSpec Student talks
20th July 2016

Institute for Computing and Information Sciences – Digital Security
Radboud University Nijmegen

A quick overview of TweetNaCl

Context

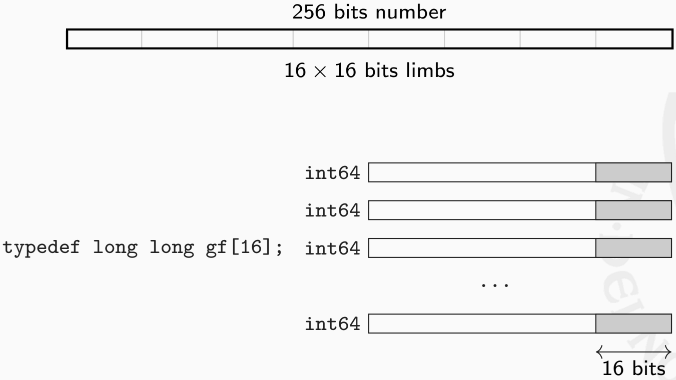
```
for(i=254;i>=0;--i) {
  r=(z[i>>3]>>(i&7))&1;
  sel25519(a,b,r);
  sel25519(c,d,r);
  A(e,a,c);               #
  Z(a,a,c);               #
  A(c,b,d);               #  Montgomery Ladder
  Z(b,b,d);               #
  S(d,e);                 #  The steps and order
  S(f,a);                 #  of the operations
  M(a,c,a);               #  have been proved
  M(c,b,e);               #  by Timmy Weerwag
  A(e,a,c);               #
  Z(a,a,c);               #
  S(b,a);                 #  The use of datatypes
  Z(c,d,f);               #  (number representation)
  M(a,c,_121665);         #  is not proven (yet).
  A(a,a,d);               #
  M(c,c,a);               #
  M(a,d,f);               #
  M(d,b,x);               #
  S(b,e);                 #
  sel25519(a,b,r);
  sel25519(c,d,r);
}
```

Code 1: crypto_scalarmult

https://tweetnacl.cr.yp.to

Datatype (or number representation)

256 bits integers does not fit into a 64 bits containers...



```
typedef long long gf[16];
```

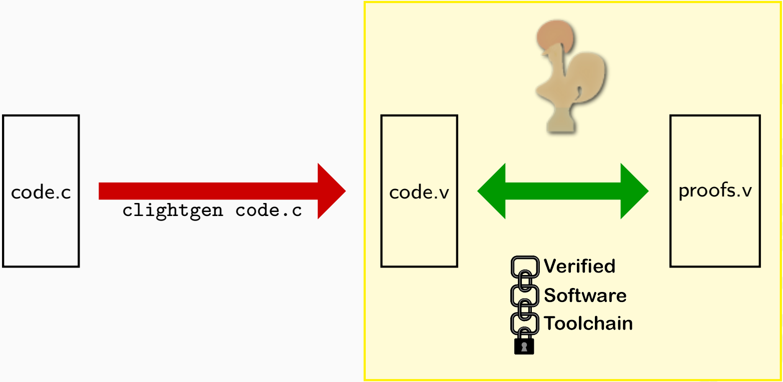
Basic Operations

```
#define FOR(i,n) for (i = 0;i < n;++i)
#define sv static void
typedef long long i64;
typedef i64 gf[16];

sv A(gf o,const gf a,const gf b)    # Addition
{
  int i;
  FOR(i,16) o[i]=a[i]+b[i];         # carrying is done separately
}

sv Z(gf o,const gf a,const gf b)    # Zubstraction
{
  int i;
  FOR(i,16) o[i]=a[i]-b[i];         # carrying is done separately
}

sv M(gf o,const gf a,const gf b)    # Multiplication
{
  i64 i,j,t[31];
  FOR(i,31) t[i]=0;
  FOR(i,16) FOR(j,16) t[i+j] = a[i]*b[j];
  FOR(i,15) t[i]+=38*t[i+16];
  FOR(i,16) o[i]=t[i];
  car25519(o);                      # carrying
  car25519(o);                      # carrying
}
```

Code 2: Basic Operations

From C to Coq

Proving with VST



code.c

clightgen code.c

code.v

proofs.v

Verified
Software
Toolchain

http://vst.cs.princeton.edu

http://compcert.inria.fr

Specification: ZofList

```
Variable n: ℤ.
Hypothesis Hn: n > 0.

(*
  in C we have gf[16] here we consider a list of integers (list ℤ)
  of length 16 in this case.

  ZofList convert a list ℤ into it's ℤ value
  assume a radix: 2^n
*)
Fixpoint ZofList (a : list ℤ) : ℤ := match a with
| [] ⇒  0
| h :: q ⇒  h + 2^n * ZofList q
end.

Notation "ℤ.lst A" := (ZofList A) (at level 65).
```

Code 3: ZofList

Example: Addition

```
Fixpoint ZsumList (a b : list ℤ) : list ℤ := match a,b with
| [], q ⇒  q
| q,[] ⇒  q
| h1::q1,h2::q2 ⇒  (Z.add h1 h2) :: ZsumList q1 q2
end.
Notation "A ⊞ B" := (ZsumList A B) (at level 60).

Corollary ZsumList_correct:
  ∀ (a b: list ℤ),
    (ℤ.lst a ⊞ b) = (ℤ.lst a) + (ℤ.lst b).
Qed.

Lemma ZsumList_bound_len:
  ∀ (m1 n1 m2 n2: ℤ) (a b: list ℤ),
    length a = length b →
    Forall (λ x ⇒  m1 < x < n1) a →
    Forall (λ x ⇒  m2 < x < n2) b →
      Forall (λ x ⇒  m1 + m2 < x < n1 + n2) (a ⊞ b).
Qed.
```

Code 4: Addition

What’s left ?

Done so far. . .

- Specification of basic operations (A,Z,M,S,Car25519).
- Bounds of basic operations.
- Proof that model matches the semantic (`code.v`) using VST.

- ∼10 months.
- compiles (`coqc`) in ∼1 hours. . . (*i7-4770K CPU @ 3.50GHz*)
- 62 lines of `C` have been verified.
- 7 180 lines of Specifications with `Coq`.
- 2 872 lines of Verification with `Coq` using VST.

Curent and future Works

- Proof of a lot of *small* utilary functions used in TweetNaCl...
- Full Proof of Montgomery Ladder's correctness.
- Proof that the model is *aligned* with Timmy's work.
- Continue on the X25519 signature scheme, Poly1305...

Thank you.