MODULE *sinsemilla*

Sinsemilla hash function spec.

Specifies the *Sinsemilla* hash function algorithm.

All variables defined represent the state of the algorithm at any given time. The algorithm is composed of a single process that hashes a message using the *Sinsemilla* hash function and predefined constants.

EXTENDS *TLC*, *Naturals*, *Integers*, *Sequences*, *Utils*, *Invariants*

CONSTANT k, c, *SinsemillaQ*, *SinsemillaS*, *Domain*, *Message*

--algorithm *sinsemilla*

variables

The bytes of the message to be hashed.

$plaintext\_bytes = CharactersToBytes(SetToSeq(Message))$,

The bytes of the domain.

$domain\_bytes = CharactersToBytes(SetToSeq(Domain))$,

The bits of the message.

$plaintext\_bits = BytesToBits(plaintext\_bytes)$,

The padded slices of the message.

$plaintext\_slices = DivideInChunks(PadBits(plaintext\_bits,\ k),\ k)$,

The point Q.

$point\_q = HashToPallas(SinsemillaQ,\ domain\_bytes)$,

The accumulator.

$accumulator = point\_q$,

The number of slices.

$n = CeilDiv(Len(plaintext\_bits),\ k)$,

The point S.

$point\_s = [a \mapsto 0,\ b \mapsto 0]$,

The bytes of the ciphertext.

$ciphertext\_bytes = \langle 0,\ 0 \rangle$

define

The type invariant.

$TypeInvariant \triangleq \land IsBytes(plaintext\_bytes) \land IsBytes(domain\_bytes) \land IsBits(plaintext\_bits)$
$\land IsSlices(plaintext\_slices) \land IsPoint(point\_q) \land IsPoint(accumulator) \land IsNumber(n)$
$\land IsPoint(point\_s) \land IsBytes(ciphertext\_bytes)$

The liveness properties.

$Liveness \triangleq \land \Diamond(accumulator \neq [a \mapsto 0,\ b \mapsto 0]) \land \Diamond(point\_s \neq [a \mapsto 0,\ b \mapsto 0])$
$\land \Diamond(plaintext\_bytes \neq ciphertext\_bytes \land ciphertext\_bytes \neq \langle 0,\ 0 \rangle)$

The safety invariants.

$Safety \triangleq \land BytesSequence(plaintext\_bytes) \land BytesSequence(domain\_bytes) \land SlicesSequence(plaintext\_sl$
$\land MaxChunks(n,\ c) \land PlainIsNotCipherText(plaintext\_bytes,\ ciphertext\_bytes)$

end define ;

The main process hash a given message using the *Sinsemilla* hash function.

fair process *main* = "MAIN"

variables $i = 1$
begin
Hash:
$point\_s := HashToPallas(SinsemillaS,\ IntToLEOSP32(plaintext\_slices[i]))$;
$accumulator := IncompleteAddition(IncompleteAddition(accumulator,\ point\_s),\ accumulator)$;
$i := i + 1$;
if $i > n$ **then**
goto *Decode* ;
else
goto *Hash* ;
end if ;
Decode the hashed point coordinates to bytes.
Decode:
$ciphertext\_bytes := \langle accumulator.a,\ accumulator.b \rangle$;
end process ;
end algorithm ;
BEGIN TRANSLATION ($chksum(pcal) =$ "$bc0d2749$" $\land\ chksum(tla) =$ "$5bbd0d5b$")
VARIABLES pc, $plaintext\_bytes$, $domain\_bytes$, $plaintext\_bits$, $plaintext\_slices$,
$point\_q$, $accumulator$, n, $point\_s$, $ciphertext\_bytes$

define statement
$TypeInvariant \triangleq \land IsBytes(plaintext\_bytes) \land IsBytes(domain\_bytes) \land IsBits(plaintext\_bits)$
$\land IsSlices(plaintext\_slices) \land IsPoint(point\_q) \land IsPoint(accumulator) \land IsNumber(n)$
$\land IsPoint(point\_s) \land IsBytes(ciphertext\_bytes)$

$Liveness \triangleq \land \Diamond(accumulator \neq [a \mapsto 0,\ b \mapsto 0]) \land \Diamond(point\_s \neq [a \mapsto 0,\ b \mapsto 0])$
$\land \Diamond(plaintext\_bytes \neq ciphertext\_bytes \land ciphertext\_bytes \neq \langle 0,\ 0 \rangle)$

$Safety \triangleq \land BytesSequence(plaintext\_bytes) \land BytesSequence(domain\_bytes) \land SlicesSequence(plaintext\_slices$
$\land MaxChunks(n,\ c) \land PlainIsNotCipherText(plaintext\_bytes,\ ciphertext\_bytes)$

VARIABLE i

$vars \triangleq \langle pc,\ plaintext\_bytes,\ domain\_bytes,\ plaintext\_bits,$
$plaintext\_slices,\ point\_q,\ accumulator,\ n,\ point\_s,$
$ciphertext\_bytes,\ i \rangle$

$ProcSet \triangleq \{$"MAIN"$\}$

$Init \triangleq$ Global variables
$\land plaintext\_bytes = CharactersToBytes(SetToSeq(Message))$
$\land domain\_bytes = CharactersToBytes(SetToSeq(Domain))$
$\land plaintext\_bits = BytesToBits(plaintext\_bytes)$
$\land plaintext\_slices = DivideInChunks(PadBits(plaintext\_bits,\ k),\ k)$
$\land point\_q = HashToPallas(SinsemillaQ,\ domain\_bytes)$
$\land accumulator = point\_q$
$\land n = CeilDiv(Len(plaintext\_bits),\ k)$

$\land\ point\_s = [a \mapsto 0,\ b \mapsto 0]$
$\land\ ciphertext\_bytes = \langle 0,\ 0 \rangle$
Process main
$\land\ i = 1$
$\land\ pc = [self \quad \in ProcSet \mapsto \text{“Hash”}]$

$Hash \triangleq$ $\land\ pc[\text{“MAIN”}] = \text{“Hash”}$
$\land\ point\_s' = HashToPallas(SinsemillaS,\ IntToLEOSP32(plaintext\_slices[i]))$
$\land\ accumulator' = IncompleteAddition(IncompleteAddition(accumulator,\ point\_s'),\ accumulator)$
$\land\ i' = i + 1$
$\land$ IF $i' > n$
THEN $\land\ pc' = [pc$ EXCEPT $![\text{“MAIN”}] = \text{“Decode”}]$
ELSE $\land\ pc' = [pc$ EXCEPT $![\text{“MAIN”}] = \text{“Hash”}]$
$\land$ UNCHANGED $\langle plaintext\_bytes,\ domain\_bytes,\ plaintext\_bits,$
$plaintext\_slices,\ point\_q,\ n,\ ciphertext\_bytes\rangle$

$Decode \triangleq$ $\land\ pc[\text{“MAIN”}] = \text{“Decode”}$
$\land\ ciphertext\_bytes' = \langle accumulator.a,\ accumulator.b\rangle$
$\land\ pc' = [pc$ EXCEPT $![\text{“MAIN”}] = \text{“Done”}]$
$\land$ UNCHANGED $\langle plaintext\_bytes,\ domain\_bytes,\ plaintext\_bits,$
$plaintext\_slices,\ point\_q,\ accumulator,\ n,\ point\_s,$
$i\rangle$

$main \triangleq Hash \lor Decode$

Allow infinite stuttering to prevent deadlock on termination.
$Terminating \triangleq$ $\land \forall\, self \in ProcSet : pc[self] = \text{“Done”}$
$\land$ UNCHANGED $vars$

$Next \triangleq$ $main$
$\lor\ Terminating$

$Spec \triangleq$ $\land\ Init \land \Box[Next]_{vars}$
$\land\ \mathrm{WF}_{vars}(main)$

$Termination \triangleq \Diamond(\forall\, self \in ProcSet : pc[self] = \text{“Done”})$

END TRANSLATION