Contents

1 cipher Theory

Built: 01 November 2017

Parent Theories: indexedLists, patternMatches

1.1 Datatypes

$$\mathit{asymMsg} = \texttt{Ea ('princ pKey) ('message option)}$$

$$\mathit{digest} = \texttt{hash ('message option)}$$

$$\mathit{pKey} = \texttt{pubK 'princ | privK 'princ}$$

$$\mathit{symKey} = \texttt{sym num}$$

$$\mathit{symMsg} = \texttt{Es symKey ('message option)}$$

1.2 Definitions

[sign_def]

$$\vdash \forall \mathit{pubKey}\ \mathit{dgst}.\ \texttt{sign}\ \mathit{pubKey}\ \mathit{dgst} = \texttt{Ea}\ \mathit{pubKey}\ (\texttt{SOME}\ \mathit{dgst})$$

[signVerify_def]

$$\begin{array}{l}
\vdash \forall \mathit{pubKey}\ \mathit{signature}\ \mathit{msgContents}. \\
\quad\quad \texttt{signVerify}\ \mathit{pubKey}\ \mathit{signature}\ \mathit{msgContents} \iff \\
\quad\quad (\texttt{SOME}\ (\texttt{hash}\ \mathit{msgContents}) = \texttt{deciphP}\ \mathit{pubKey}\ \mathit{signature})
\end{array}$$

1.3 Theorems

[asymMsg_one_one]

$$\begin{array}{l}
\vdash \forall a_0\ a_1\ a_0'\ a_1'. \\
\quad\quad (\texttt{Ea}\ a_0\ a_1 = \texttt{Ea}\ a_0'\ a_1') \iff (a_0 = a_0') \land (a_1 = a_1')
\end{array}$$

[deciphP_clauses]

$$\begin{array}{l}
\vdash (\forall P\ \mathit{text}. \\
\quad\quad\quad (\texttt{deciphP}\ (\texttt{pubK}\ P)\ (\texttt{Ea}\ (\texttt{privK}\ P)\ (\texttt{SOME}\ \mathit{text})) = \\
\quad\quad\quad\ \ \texttt{SOME}\ \mathit{text}) \land \\
\quad\quad\quad (\texttt{deciphP}\ (\texttt{privK}\ P)\ (\texttt{Ea}\ (\texttt{pubK}\ P)\ (\texttt{SOME}\ \mathit{text})) = \\
\quad\quad\quad\ \ \texttt{SOME}\ \mathit{text})) \land \\
\quad (\forall k\ P\ \mathit{text}. \\
\quad\quad\quad (\texttt{deciphP}\ k\ (\texttt{Ea}\ (\texttt{privK}\ P)\ (\texttt{SOME}\ \mathit{text})) = \texttt{SOME}\ \mathit{text}) \iff \\
\quad\quad\quad (k = \texttt{pubK}\ P)) \land \\
\quad (\forall k\ P\ \mathit{text}. \\
\quad\quad\quad (\texttt{deciphP}\ k\ (\texttt{Ea}\ (\texttt{pubK}\ P)\ (\texttt{SOME}\ \mathit{text})) = \texttt{SOME}\ \mathit{text}) \iff \\
\quad\quad\quad (k = \texttt{privK}\ P)) \land \\
\quad (\forall x\ k_2\ k_1\ P_2\ P_1. \\
\quad\quad\quad (\texttt{deciphP}\ (\texttt{pubK}\ P_1)\ (\texttt{Ea}\ (\texttt{pubK}\ P_2)\ (\texttt{SOME}\ x)) = \texttt{NONE}) \land \\
\quad\quad\quad (\texttt{deciphP}\ k_1\ (\texttt{Ea}\ k_2\ \texttt{NONE}) = \texttt{NONE})) \land \\
\quad \forall x\ P_2\ P_1.\ \texttt{deciphP}\ (\texttt{privK}\ P_1)\ (\texttt{Ea}\ (\texttt{privK}\ P_2)\ (\texttt{SOME}\ x)) = \texttt{NONE}
\end{array}$$

[deciphP_def]

```
⊢ (deciphP key (Ea (privK P) (SOME x)) =
   if key = pubK P then SOME x else NONE) ∧
  (deciphP key (Ea (pubK P) (SOME x)) =
   if key = privK P then SOME x else NONE) ∧
  (deciphP k1 (Ea k2 NONE) = NONE)
```

[deciphP_ind]

```
⊢ ∀P'.
    (∀key P x. P' key (Ea (privK P) (SOME x))) ∧
    (∀key P x. P' key (Ea (pubK P) (SOME x))) ∧
    (∀k1 k2. P' k1 (Ea k2 NONE)) ⇒
    ∀v v1. P' v v1
```

[deciphP_one_one]

```
⊢ (∀P1 P2 text1 text2.
     (deciphP (pubK P1) (Ea (privK P2) (SOME text2)) =
      SOME text1) ⇔ (P1 = P2) ∧ (text1 = text2)) ∧
  (∀P1 P2 text1 text2.
     (deciphP (privK P1) (Ea (pubK P2) (SOME text2)) =
      SOME text1) ⇔ (P1 = P2) ∧ (text1 = text2)) ∧
  (∀p c P msg.
     (deciphP (pubK P) (Ea p c) = SOME msg) ⇔
     (p = privK P) ∧ (c = SOME msg)) ∧
  (∀enMsg P msg.
     (deciphP (pubK P) enMsg = SOME msg) ⇔
     (enMsg = Ea (privK P) (SOME msg))) ∧
  (∀p c P msg.
     (deciphP (privK P) (Ea p c) = SOME msg) ⇔
     (p = pubK P) ∧ (c = SOME msg)) ∧
  ∀enMsg P msg.
    (deciphP (privK P) enMsg = SOME msg) ⇔
    (enMsg = Ea (pubK P) (SOME msg))
```

[deciphS_clauses]

```
⊢ (∀k text. deciphS k (Es k (SOME text)) = SOME text) ∧
  (∀k1 k2 text.
     (deciphS k1 (Es k2 (SOME text)) = SOME text) ⇔
     (k1 = k2)) ∧
  (∀k1 k2 text.
     (deciphS k1 (Es k2 (SOME text)) = NONE) ⇔ k1 ≠ k2) ∧
  ∀k1 k2. deciphS k1 (Es k2 NONE) = NONE
```

[deciphS_def]

```
⊢ (deciphS k1 (Es k2 (SOME x)) =
   if k1 = k2 then SOME x else NONE) ∧
  (deciphS k1 (Es k2 NONE) = NONE)
```

[deciphS_ind]

$$\begin{array}{l}\vdash \forall P.\\ \quad (\forall k_1\ k_2\ x.\ P\ k_1\ (\texttt{Es}\ k_2\ (\texttt{SOME}\ x))) \land\\ \quad (\forall k_1\ k_2.\ P\ k_1\ (\texttt{Es}\ k_2\ \texttt{NONE})) \Rightarrow\\ \quad \forall v\ v_1.\ P\ v\ v_1\end{array}$$

[deciphS_one_one]

$$\begin{array}{l}\vdash (\forall k_1\ k_2\ \mathit{text}_1\ \mathit{text}_2.\\ \qquad (\texttt{deciphS}\ k_1\ (\texttt{Es}\ k_2\ (\texttt{SOME}\ \mathit{text}_2)) = \texttt{SOME}\ \mathit{text}_1) \iff\\ \qquad (k_1 = k_2) \land (\mathit{text}_1 = \mathit{text}_2)) \land\\ \quad \forall \mathit{enMsg}\ \mathit{text}\ \mathit{key}.\\ \qquad (\texttt{deciphS}\ \mathit{key}\ \mathit{enMsg} = \texttt{SOME}\ \mathit{text}) \iff\\ \qquad (\mathit{enMsg} = \texttt{Es}\ \mathit{key}\ (\texttt{SOME}\ \mathit{text}))\end{array}$$

[digest_one_one]

$$\vdash \forall a\ a'.\ (\texttt{hash}\ a = \texttt{hash}\ a') \iff (a = a')$$

[option_distinct]

$$\vdash \forall x.\ \texttt{NONE} \neq \texttt{SOME}\ x$$

[option_one_one]

$$\vdash \forall x\ y.\ (\texttt{SOME}\ x = \texttt{SOME}\ y) \iff (x = y)$$

[pKey_distinct_clauses]

$$\vdash (\forall a'\ a.\ \texttt{pubK}\ a \neq \texttt{privK}\ a') \land \forall a'\ a.\ \texttt{privK}\ a' \neq \texttt{pubK}\ a$$

[pKey_one_one]

$$\begin{array}{l}\vdash (\forall a\ a'.\ (\texttt{pubK}\ a = \texttt{pubK}\ a') \iff (a = a')) \land\\ \quad \forall a\ a'.\ (\texttt{privK}\ a = \texttt{privK}\ a') \iff (a = a')\end{array}$$

[sign_one_one]

$$\begin{array}{l}\vdash \forall \mathit{pubKey}_1\ \mathit{pubKey}_2\ m_1\ m_2.\\ \quad (\texttt{sign}\ \mathit{pubKey}_1\ (\texttt{hash}\ m_1) = \texttt{sign}\ \mathit{pubKey}_2\ (\texttt{hash}\ m_2)) \iff\\ \quad (\mathit{pubKey}_1 = \mathit{pubKey}_2) \land (m_1 = m_2)\end{array}$$

[signVerify_one_one]

$$\begin{array}{l}\vdash (\forall P\ m_1\ m_2.\\ \qquad \texttt{signVerify}\ (\texttt{pubK}\ P)\ (\texttt{Ea}\ (\texttt{privK}\ P)\ (\texttt{SOME}\ (\texttt{hash}\ (\texttt{SOME}\ m_1))))\\ \qquad\quad (\texttt{SOME}\ m_2) \iff (m_1 = m_2)) \land\\ \quad (\forall \mathit{signature}\ P\ \mathit{text}.\\ \qquad \texttt{signVerify}\ (\texttt{pubK}\ P)\ \mathit{signature}\ (\texttt{SOME}\ \mathit{text}) \iff\\ \qquad (\mathit{signature} = \texttt{sign}\ (\texttt{privK}\ P)\ (\texttt{hash}\ (\texttt{SOME}\ \mathit{text})))) \land\\ \quad \forall \mathit{text}_2\ \mathit{text}_1\ P_2\ P_1.\\ \qquad \texttt{signVerify}\ (\texttt{pubK}\ P_1)\ (\texttt{sign}\ (\texttt{privK}\ P_2)\ (\texttt{hash}\ (\texttt{SOME}\ \mathit{text}_2)))\\ \qquad\quad (\texttt{SOME}\ \mathit{text}_1) \iff (P_1 = P_2) \land (\mathit{text}_1 = \mathit{text}_2)\end{array}$$

[signVerifyOK]

```
⊢ ∀P msg.
    signVerify (pubK P) (sign (privK P) (hash (SOME msg)))
      (SOME msg)
```

[symKey_one_one]

```
⊢ ∀a a'. (sym a = sym a') ⇔ (a = a')
```

[symMsg_one_one]

```
⊢ ∀a0 a1 a0' a1'.
    (Es a0 a1 = Es a0' a1') ⇔ (a0 = a0') ∧ (a1 = a1')
```

2 cryptoExercises Theory

Built: 01 November 2017

Parent Theories: cipher, string

2.1 Theorems

[exercise15_6_1a_thm]

```
⊢ ∀key enMsg message.
    (deciphS key enMsg = SOME message) ⇔
    (enMsg = Es key (SOME message))
```

[exercise15_6_1b_thm]

```
⊢ ∀keyAlice k text.
    (deciphS keyAlice (Es k (SOME text)) =
     SOME “This is from Alice”) ⇔
    (k = keyAlice) ∧ (text = “This is from Alice”)
```

[exercise15_6_2a_thm]

```
⊢ ∀P message.
    (deciphP (pubK P) enMsg = SOME message) ⇔
    (enMsg = Ea (privK P) (SOME message))
```

[exercise15_6_2b_thm]

```
⊢ ∀key text.
    (deciphP (pubK Alice) (Ea key (SOME text)) =
     SOME “This is from Alice”) ⇔
    (key = privK Alice) ∧ (text = “This is from Alice”)
```

[exercise15_6_3_thm]

```
⊢ ∀signature.
    signVerify (pubK Alice) signature
      (SOME “This is from Alice”) ⇔
    (signature =
     sign (privK Alice) (hash (SOME “This is from Alice”)))
```

Index