Contents

1 cryptoExercises Theory

Built: 31 August 2019

Parent Theories: cipher, string

1.1 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”)))
```

2 cipher Theory

Built: 31 August 2019

Parent Theories: indexedLists, patternMatches

2.1 Datatypes

```
asymMsg = Ea ('princ pKey) ('message option)

digest = hash ('message option)

pKey = pubK 'princ | privK 'princ

symKey = sym num

symMsg = Es symKey ('message option)
```

2.2 Definitions

[sign_def]

```
⊢ ∀ pubKey dgst. sign pubKey dgst = Ea pubKey (SOME dgst)
```

[signVerify_def]

```
⊢ ∀ pubKey signature msgContents.
    signVerify pubKey signature msgContents ⇔
    (SOME (hash msgContents) = deciphP pubKey signature)
```

2.3 Theorems

[asymMsg_one_one]

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

[deciphP_clauses]

```
⊢ (∀ P text.
      (deciphP (pubK P) (Ea (privK P) (SOME text)) =
       SOME text) ∧
      (deciphP (privK P) (Ea (pubK P) (SOME text)) =
       SOME text)) ∧
  (∀ k P text.
      (deciphP k (Ea (privK P) (SOME text)) = SOME text) ⇔
      (k = pubK P)) ∧
  (∀ k P text.
      (deciphP k (Ea (pubK P) (SOME text)) = SOME text) ⇔
      (k = privK P)) ∧
  (∀ x k2 k1 P2 P1.
      (deciphP (pubK P1) (Ea (pubK P2) (SOME x)) = NONE) ∧
      (deciphP k1 (Ea k2 NONE) = NONE)) ∧
  ∀ x P2 P1. deciphP (privK P1) (Ea (privK P2) (SOME x)) = NONE
```

[deciphP_def]

$$
\begin{array}{l}
\vdash (\texttt{deciphP}\ \mathit{key}\ (\texttt{Ea}\ (\texttt{privK}\ P)\ (\texttt{SOME}\ x)) = \\
\quad \textbf{if}\ \mathit{key} = \texttt{pubK}\ P\ \textbf{then}\ \texttt{SOME}\ x\ \textbf{else}\ \texttt{NONE}) \land \\
\quad (\texttt{deciphP}\ \mathit{key}\ (\texttt{Ea}\ (\texttt{pubK}\ P)\ (\texttt{SOME}\ x)) = \\
\quad \textbf{if}\ \mathit{key} = \texttt{privK}\ P\ \textbf{then}\ \texttt{SOME}\ x\ \textbf{else}\ \texttt{NONE}) \land \\
\quad (\texttt{deciphP}\ k_1\ (\texttt{Ea}\ k_2\ \texttt{NONE}) = \texttt{NONE})
\end{array}
$$

[deciphP_ind]

$$
\begin{array}{l}
\vdash \forall P'. \\
\qquad (\forall \mathit{key}\ P\ x.\ P'\ \mathit{key}\ (\texttt{Ea}\ (\texttt{privK}\ P)\ (\texttt{SOME}\ x))) \land \\
\qquad (\forall \mathit{key}\ P\ x.\ P'\ \mathit{key}\ (\texttt{Ea}\ (\texttt{pubK}\ P)\ (\texttt{SOME}\ x))) \land \\
\qquad (\forall k_1\ k_2.\ P'\ k_1\ (\texttt{Ea}\ k_2\ \texttt{NONE})) \Rightarrow \\
\qquad \forall v\ v_1.\ P'\ v\ v_1
\end{array}
$$

[deciphP_one_one]

$$
\begin{array}{l}
\vdash (\forall P_1\ P_2\ \mathit{text}_1\ \mathit{text}_2. \\
\qquad (\texttt{deciphP}\ (\texttt{pubK}\ P_1)\ (\texttt{Ea}\ (\texttt{privK}\ P_2)\ (\texttt{SOME}\ \mathit{text}_2)) = \\
\qquad\ \texttt{SOME}\ \mathit{text}_1) \iff (P_1 = P_2) \land (\mathit{text}_1 = \mathit{text}_2)) \land \\
\quad (\forall P_1\ P_2\ \mathit{text}_1\ \mathit{text}_2. \\
\qquad (\texttt{deciphP}\ (\texttt{privK}\ P_1)\ (\texttt{Ea}\ (\texttt{pubK}\ P_2)\ (\texttt{SOME}\ \mathit{text}_2)) = \\
\qquad\ \texttt{SOME}\ \mathit{text}_1) \iff (P_1 = P_2) \land (\mathit{text}_1 = \mathit{text}_2)) \land \\
\quad (\forall p\ c\ P\ \mathit{msg}. \\
\qquad (\texttt{deciphP}\ (\texttt{pubK}\ P)\ (\texttt{Ea}\ p\ c) = \texttt{SOME}\ \mathit{msg}) \iff \\
\qquad (p = \texttt{privK}\ P) \land (c = \texttt{SOME}\ \mathit{msg})) \land \\
\quad (\forall \mathit{enMsg}\ P\ \mathit{msg}. \\
\qquad (\texttt{deciphP}\ (\texttt{pubK}\ P)\ \mathit{enMsg} = \texttt{SOME}\ \mathit{msg}) \iff \\
\qquad (\mathit{enMsg} = \texttt{Ea}\ (\texttt{privK}\ P)\ (\texttt{SOME}\ \mathit{msg}))) \land \\
\quad (\forall p\ c\ P\ \mathit{msg}. \\
\qquad (\texttt{deciphP}\ (\texttt{privK}\ P)\ (\texttt{Ea}\ p\ c) = \texttt{SOME}\ \mathit{msg}) \iff \\
\qquad (p = \texttt{pubK}\ P) \land (c = \texttt{SOME}\ \mathit{msg})) \land \\
\quad \forall \mathit{enMsg}\ P\ \mathit{msg}. \\
\qquad (\texttt{deciphP}\ (\texttt{privK}\ P)\ \mathit{enMsg} = \texttt{SOME}\ \mathit{msg}) \iff \\
\qquad (\mathit{enMsg} = \texttt{Ea}\ (\texttt{pubK}\ P)\ (\texttt{SOME}\ \mathit{msg}))
\end{array}
$$

[deciphS_clauses]

$$
\begin{array}{l}
\vdash (\forall k\ \mathit{text}.\ \texttt{deciphS}\ k\ (\texttt{Es}\ k\ (\texttt{SOME}\ \mathit{text})) = \texttt{SOME}\ \mathit{text}) \land \\
\quad (\forall k_1\ k_2\ \mathit{text}. \\
\qquad (\texttt{deciphS}\ k_1\ (\texttt{Es}\ k_2\ (\texttt{SOME}\ \mathit{text})) = \texttt{SOME}\ \mathit{text}) \iff \\
\qquad (k_1 = k_2)) \land \\
\quad (\forall k_1\ k_2\ \mathit{text}. \\
\qquad (\texttt{deciphS}\ k_1\ (\texttt{Es}\ k_2\ (\texttt{SOME}\ \mathit{text})) = \texttt{NONE}) \iff k_1 \neq k_2) \land \\
\quad \forall k_1\ k_2.\ \texttt{deciphS}\ k_1\ (\texttt{Es}\ k_2\ \texttt{NONE}) = \texttt{NONE}
\end{array}
$$

[deciphS_def]

$$\begin{array}{l}
\vdash (\mathtt{deciphS}\ k_1\ (\mathtt{Es}\ k_2\ (\mathtt{SOME}\ x)) = \\
\quad \textbf{if}\ k_1 = k_2\ \textbf{then}\ \mathtt{SOME}\ x\ \textbf{else}\ \mathtt{NONE}) \land \\
\quad (\mathtt{deciphS}\ k_1\ (\mathtt{Es}\ k_2\ \mathtt{NONE}) = \mathtt{NONE})
\end{array}$$

[deciphS_ind]

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

[deciphS_one_one]

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

[digest_one_one]

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

[option_distinct]

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

[option_one_one]

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

[pKey_distinct_clauses]

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

[pKey_one_one]

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

[sign_one_one]

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

[signVerify_one_one]

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

[signVerifyOK]

$$\begin{array}{l}
\vdash \forall P\ \mathit{msg} .\\
\quad\quad \mathtt{signVerify}\ (\mathtt{pubK}\ P)\ (\mathtt{sign}\ (\mathtt{privK}\ P)\ (\mathtt{hash}\ (\mathtt{SOME}\ \mathit{msg})))\\
\quad\quad\quad (\mathtt{SOME}\ \mathit{msg})
\end{array}$$

[symKey_one_one]

$$\vdash \forall a\ a' .\ (\mathtt{sym}\ a = \mathtt{sym}\ a') \iff (a = a')$$

[symMsg_one_one]

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

Index