

Contents

1	cipher Theory	3
1.1	Datatypes	3
1.2	Definitions	3
1.3	Theorems	3
2	cryptoExercises Theory	6
2.1	Theorems	6

1 cipher Theory

Built: 26 March 2019

Parent Theories: indexedLists, patternMatches

1.1 Datatypes

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

digest = hash ('message option)

pKey = pubK 'princ | privK 'princ

symKey = sym num

symMsg = Es symKey ('message option)

1.2 Definitions

[sign_def]

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

[signVerify_def]

$\vdash \forall \text{pubKey } \text{signature } \text{msgContents}.$
 $\text{signVerify } \text{pubKey } \text{signature } \text{msgContents} \iff$
 $(\text{SOME } (\text{hash } \text{msgContents}) = \text{deciphP } \text{pubKey } \text{signature})$

1.3 Theorems

[asymMsg_one_one]

$\vdash \forall a_0 \ a_1 \ a'_0 \ a'_1.$
 $(\text{Ea } a_0 \ a_1 = \text{Ea } a'_0 \ a'_1) \iff (a_0 = a'_0) \wedge (a_1 = a'_1)$

[deciphP_clauses]

$\vdash (\forall P \ \text{text}.$
 $(\text{deciphP } (\text{pubK } P) (\text{Ea } (\text{privK } P) (\text{SOME } \text{text}))) =$
 $\text{SOME } \text{text}) \wedge$
 $(\text{deciphP } (\text{privK } P) (\text{Ea } (\text{pubK } P) (\text{SOME } \text{text}))) =$
 $\text{SOME } \text{text})) \wedge$
 $(\forall k \ P \ \text{text}.$
 $(\text{deciphP } k (\text{Ea } (\text{privK } P) (\text{SOME } \text{text}))) = \text{SOME } \text{text}) \iff$
 $(k = \text{pubK } P)) \wedge$
 $(\forall k \ P \ \text{text}.$

$$\begin{aligned}
& (\text{deciphP } k \text{ (Ea (pubK } P) \text{ (SOME } \textit{text}))} = \text{SOME } \textit{text}) \iff \\
& (k = \text{privK } P) \wedge \\
& (\forall x \ k_2 \ k_1 \ P_2 \ P_1. \\
& \quad (\text{deciphP (pubK } P_1) \text{ (Ea (pubK } P_2) \text{ (SOME } x))} = \text{NONE}) \wedge \\
& \quad (\text{deciphP } k_1 \text{ (Ea } k_2 \text{ NONE)} = \text{NONE})) \wedge \\
& \forall x \ P_2 \ P_1. \text{deciphP (privK } P_1) \text{ (Ea (privK } P_2) \text{ (SOME } x))} = \text{NONE}
\end{aligned}$$

[deciphP_def]

$$\begin{aligned}
& \vdash (\text{deciphP } \textit{key} \text{ (Ea (privK } P) \text{ (SOME } x))} = \\
& \quad \text{if } \textit{key} = \text{pubK } P \text{ then SOME } x \text{ else NONE}) \wedge \\
& (\text{deciphP } \textit{key} \text{ (Ea (pubK } P) \text{ (SOME } x))} = \\
& \quad \text{if } \textit{key} = \text{privK } P \text{ then SOME } x \text{ else NONE}) \wedge \\
& (\text{deciphP } k_1 \text{ (Ea } k_2 \text{ NONE)} = \text{NONE})
\end{aligned}$$

[deciphP_ind]

$$\begin{aligned}
& \vdash \forall P'. \\
& \quad (\forall \textit{key} \ P \ x. \ P' \ \textit{key} \text{ (Ea (privK } P) \text{ (SOME } x))) \wedge \\
& \quad (\forall \textit{key} \ P \ x. \ P' \ \textit{key} \text{ (Ea (pubK } P) \text{ (SOME } x))) \wedge \\
& \quad (\forall k_1 \ k_2. \ P' \ k_1 \text{ (Ea } k_2 \text{ NONE)}) \Rightarrow \\
& \quad \forall v \ v_1. \ P' \ v \ v_1
\end{aligned}$$

[deciphP_one_one]

$$\begin{aligned}
& \vdash (\forall P_1 \ P_2 \ \textit{text}_1 \ \textit{text}_2. \\
& \quad (\text{deciphP (pubK } P_1) \text{ (Ea (privK } P_2) \text{ (SOME } \textit{text}_2))} = \\
& \quad \text{SOME } \textit{text}_1) \iff (P_1 = P_2) \wedge (\textit{text}_1 = \textit{text}_2)) \wedge \\
& (\forall P_1 \ P_2 \ \textit{text}_1 \ \textit{text}_2. \\
& \quad (\text{deciphP (privK } P_1) \text{ (Ea (pubK } P_2) \text{ (SOME } \textit{text}_2))} = \\
& \quad \text{SOME } \textit{text}_1) \iff (P_1 = P_2) \wedge (\textit{text}_1 = \textit{text}_2)) \wedge \\
& (\forall p \ c \ P \ \textit{msg}. \\
& \quad (\text{deciphP (pubK } P) \text{ (Ea } p \ c) = \text{SOME } \textit{msg}) \iff \\
& \quad (p = \text{privK } P) \wedge (c = \text{SOME } \textit{msg})) \wedge \\
& (\forall \textit{enMsg} \ P \ \textit{msg}. \\
& \quad (\text{deciphP (pubK } P) \ \textit{enMsg} = \text{SOME } \textit{msg}) \iff \\
& \quad (\textit{enMsg} = \text{Ea (privK } P) \text{ (SOME } \textit{msg}))) \wedge \\
& (\forall p \ c \ P \ \textit{msg}. \\
& \quad (\text{deciphP (privK } P) \text{ (Ea } p \ c) = \text{SOME } \textit{msg}) \iff \\
& \quad (p = \text{pubK } P) \wedge (c = \text{SOME } \textit{msg})) \wedge \\
& \forall \textit{enMsg} \ P \ \textit{msg}. \\
& \quad (\text{deciphP (privK } P) \ \textit{enMsg} = \text{SOME } \textit{msg}) \iff \\
& \quad (\textit{enMsg} = \text{Ea (pubK } P) \text{ (SOME } \textit{msg}))
\end{aligned}$$

[deciphS_clauses]

$$\begin{aligned}
& \vdash (\forall k \ \textit{text}. \text{deciphS } k \text{ (Es } k \text{ (SOME } \textit{text}))} = \text{SOME } \textit{text}) \wedge \\
& (\forall k_1 \ k_2 \ \textit{text}.
\end{aligned}$$

$$\begin{aligned}
& (\text{deciphS } k_1 \text{ (Es } k_2 \text{ (SOME } \textit{text}))} = \text{SOME } \textit{text}) \iff \\
& (k_1 = k_2) \wedge \\
& (\forall k_1 \ k_2 \ \textit{text}. \\
& \quad (\text{deciphS } k_1 \text{ (Es } k_2 \text{ (SOME } \textit{text}))} = \text{NONE}) \iff k_1 \neq k_2) \wedge \\
& \quad \forall k_1 \ k_2. \text{deciphS } k_1 \text{ (Es } k_2 \text{ NONE)} = \text{NONE}
\end{aligned}$$

[deciphS_def]

$$\begin{aligned}
& \vdash (\text{deciphS } k_1 \text{ (Es } k_2 \text{ (SOME } x))} = \\
& \quad \text{if } k_1 = k_2 \text{ then SOME } x \text{ else NONE}) \wedge \\
& \quad (\text{deciphS } k_1 \text{ (Es } k_2 \text{ NONE)} = \text{NONE})
\end{aligned}$$

[deciphS_ind]

$$\begin{aligned}
& \vdash \forall P. \\
& \quad (\forall k_1 \ k_2 \ x. P \ k_1 \text{ (Es } k_2 \text{ (SOME } x))) \wedge \\
& \quad (\forall k_1 \ k_2. P \ k_1 \text{ (Es } k_2 \text{ NONE)}) \Rightarrow \\
& \quad \forall v \ v_1. P \ v \ v_1
\end{aligned}$$

[deciphS_one_one]

$$\begin{aligned}
& \vdash (\forall k_1 \ k_2 \ \textit{text}_1 \ \textit{text}_2. \\
& \quad (\text{deciphS } k_1 \text{ (Es } k_2 \text{ (SOME } \textit{text}_2))} = \text{SOME } \textit{text}_1) \iff \\
& \quad (k_1 = k_2) \wedge (\textit{text}_1 = \textit{text}_2)) \wedge \\
& \quad \forall \textit{enMsg} \ \textit{key}. \\
& \quad (\text{deciphS } \textit{key} \ \textit{enMsg} = \text{SOME } \textit{text}) \iff \\
& \quad (\textit{enMsg} = \text{Es } \textit{key} \text{ (SOME } \textit{text}))
\end{aligned}$$

[digest_one_one]

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

[option_distinct]

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

[option_one_one]

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

[pKey_distinct_clauses]

$$\vdash (\forall a' \ a. \text{pubK } a \neq \text{privK } a') \wedge \forall a' \ a. \text{privK } a' \neq \text{pubK } a$$

[pKey_one_one]

$$\begin{aligned}
& \vdash (\forall a \ a'. (\text{pubK } a = \text{pubK } a') \iff (a = a')) \wedge \\
& \quad \forall a \ a'. (\text{privK } a = \text{privK } a') \iff (a = a')
\end{aligned}$$

[sign_one_one]

$$\vdash \forall \text{pubKey}_1 \text{ pubKey}_2 m_1 m_2. \\ (\text{sign } \text{pubKey}_1 (\text{hash } m_1) = \text{sign } \text{pubKey}_2 (\text{hash } m_2)) \iff \\ (\text{pubKey}_1 = \text{pubKey}_2) \wedge (m_1 = m_2)$$

[signVerify_one_one]

$$\vdash (\forall P m_1 m_2. \\ \text{signVerify } (\text{pubK } P) (\text{Ea } (\text{privK } P) (\text{SOME } (\text{hash } (\text{SOME } m_1)))) \\ (\text{SOME } m_2) \iff (m_1 = m_2)) \wedge \\ (\forall \text{signature } P \text{ text}. \\ \text{signVerify } (\text{pubK } P) \text{ signature } (\text{SOME } \text{text}) \iff \\ (\text{signature} = \text{sign } (\text{privK } P) (\text{hash } (\text{SOME } \text{text})))) \wedge \\ \forall \text{text}_2 \text{ text}_1 P_2 P_1. \\ \text{signVerify } (\text{pubK } P_1) (\text{sign } (\text{privK } P_2) (\text{hash } (\text{SOME } \text{text}_2))) \\ (\text{SOME } \text{text}_1) \iff (P_1 = P_2) \wedge (\text{text}_1 = \text{text}_2))$$

[signVerifyOK]

$$\vdash \forall P \text{ msg}. \\ \text{signVerify } (\text{pubK } P) (\text{sign } (\text{privK } P) (\text{hash } (\text{SOME } \text{msg}))) \\ (\text{SOME } \text{msg})$$

[symKey_one_one]

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

[symMsg_one_one]

$$\vdash \forall a_0 a_1 a'_0 a'_1. \\ (\text{Es } a_0 a_1 = \text{Es } a'_0 a'_1) \iff (a_0 = a'_0) \wedge (a_1 = a'_1)$$

2 cryptoExercises Theory

Built: 26 March 2019

Parent Theories: cipher, string

2.1 Theorems

[exercise15_6_1a_thm]

$$\vdash \forall \text{key } \text{enMsg } \text{message}. \\ (\text{deciphS } \text{key } \text{enMsg} = \text{SOME } \text{message}) \iff \\ (\text{enMsg} = \text{Es } \text{key } (\text{SOME } \text{message}))$$

[exercise15_6_1b_thm]

$$\begin{aligned} &\vdash \forall \text{keyAlice } k \text{ text.} \\ &\quad (\text{deciphS } \text{keyAlice } (\text{Es } k \text{ (SOME } \text{text})) = \\ &\quad \text{SOME "This is from Alice"}) \iff \\ &\quad (k = \text{keyAlice}) \wedge (\text{text} = \text{"This is from Alice"}) \end{aligned}$$

[exercise15_6_2a_thm]

$$\begin{aligned} &\vdash \forall P \text{ message.} \\ &\quad (\text{deciphP } (\text{pubK } P) \text{ enMsg} = \text{SOME } \text{message}) \iff \\ &\quad (\text{enMsg} = \text{Ea } (\text{privK } P) (\text{SOME } \text{message})) \end{aligned}$$

[exercise15_6_2b_thm]

$$\begin{aligned} &\vdash \forall \text{key } \text{text.} \\ &\quad (\text{deciphP } (\text{pubK } \text{Alice}) (\text{Ea } \text{key } (\text{SOME } \text{text})) = \\ &\quad \text{SOME "This is from Alice"}) \iff \\ &\quad (\text{key} = \text{privK Alice}) \wedge (\text{text} = \text{"This is from Alice"}) \end{aligned}$$

[exercise15_6_3_thm]

$$\begin{aligned} &\vdash \forall \text{signature.} \\ &\quad \text{signVerify } (\text{pubK } \text{Alice}) \text{ signature} \\ &\quad (\text{SOME "This is from Alice"}) \iff \\ &\quad (\text{signature} = \\ &\quad \text{sign } (\text{privK } \text{Alice}) (\text{hash } (\text{SOME "This is from Alice"}))) \end{aligned}$$

Index

cipher Theory, 3

- Datatypes, 3
- Definitions, 3
 - sign_def, 3
 - signVerify_def, 3
- Theorems, 3
 - asymMsg_one_one, 3
 - deciphP_clauses, 3
 - deciphP_def, 4
 - deciphP_ind, 4
 - deciphP_one_one, 4
 - deciphS_clauses, 4
 - deciphS_def, 5
 - deciphS_ind, 5
 - deciphS_one_one, 5
 - digest_one_one, 5
 - option_distinct, 5
 - option_one_one, 5
 - pKey_distinct_clauses, 5
 - pKey_one_one, 5
 - sign_one_one, 6
 - signVerify_one_one, 6
 - signVerifyOK, 6
 - symKey_one_one, 6
 - symMsg_one_one, 6

cryptoExercises Theory, 6

- Theorems, 6
 - exercise15.6.1a.thm, 6
 - exercise15.6.1b.thm, 7
 - exercise15.6.2a.thm, 7
 - exercise15.6.2b.thm, 7
 - exercise15.6.3.thm, 7