

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: 07 March 2020

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](#)]

$\vdash (\forall P \ \text{text}.$
 $(\text{deciphP } (\text{pubK } P) (\text{Ea } (\text{privK } P) (\text{SOME } \text{text}))) =$
 $\text{SOME } \text{text}) \wedge$
 $(\text{deciphP } (\text{privK } P) (\text{Ea } (\text{pubK } P) (\text{SOME } \text{text}))) =$
 $\text{SOME } \text{text})) \wedge$
 $(\forall k \ P \ \text{text}.$
 $(\text{deciphP } k (\text{Ea } (\text{privK } P) (\text{SOME } \text{text}))) = \text{SOME } \text{text}) \iff$
 $(k = \text{pubK } P)) \wedge$
 $(\forall k \ P \ \text{text}.$
 $(\text{deciphP } k (\text{Ea } (\text{pubK } P) (\text{SOME } \text{text}))) = \text{SOME } \text{text}) \iff$
 $(k = \text{privK } P)) \wedge$
 $(\forall x \ k_2 \ k_1 \ P_2 \ P_1.$
 $(\text{deciphP } (\text{pubK } P_1) (\text{Ea } (\text{pubK } P_2) (\text{SOME } x))) = \text{NONE}) \wedge$
 $(\text{deciphP } k_1 (\text{Ea } k_2 \ \text{NONE}) = \text{NONE})) \wedge$
 $\forall x \ P_2 \ P_1. \text{deciphP } (\text{privK } P_1) (\text{Ea } (\text{privK } P_2) (\text{SOME } x)) = \text{NONE}$

[deciphP_def]

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

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

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

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

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

[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]

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

$$\begin{aligned} &\vdash \forall \text{ pubKey}_1 \text{ pubKey}_2 m_1 m_2. \\ &\quad (\text{sign } \text{ pubKey}_1 (\text{hash } m_1) = \text{sign } \text{ pubKey}_2 (\text{hash } m_2)) \iff \\ &\quad (\text{pubKey}_1 = \text{pubKey}_2) \wedge (m_1 = m_2) \end{aligned}$$
[signVerify_one_one]

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

[signVerifyOK]

$\vdash \forall P \text{ msg.}$
 $\text{signVerify (pubK } P) (\text{sign (privK } P) (\text{hash (SOME msg)}))$
 (SOME 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: 07 March 2020

Parent Theories: cipher, string

2.1 Theorems

[exercise15_6_1a_thm]

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

[exercise15_6_1b_thm]

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

[exercise15_6_2a_thm]

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

[exercise15_6_2b_thm]

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

[exercise15_6_3_thm]

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

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, 4

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, 5

signVerify_one_one, 5

signVerifyOK, 6

symKey_one_one, 6

symMsg_one_one, 6

cryptoExercises Theory, 6

Theorems, 6

exercise15_6_1a_thm, 6

exercise15_6_1b_thm, 6

exercise15_6_2a_thm, 6

exercise15_6_2b_thm, 6

exercise15_6_3_thm, 6