DATE: 2/18/15 6.857 Hash functions PAGE: LOS. 1 Pset #1 due Mon 2/23. [Submiteuch problem as separate pdf.] Admin: Pset #2 out Mon 2/23. Charles River Crypto Day: Friday 2/20 (does everyone have access to secret student area?) Cryptographic Hash Functions Today: · definition · random oracle model (ROM) · desirable properties (CR, OW, ... · applications e construction Readings: Katz/Lindell Chapter 5 Chapter 11 Part/Pelz1 Chapter 5 Ferguson Wikipedia SHA-3

								1	MAG	E.	Q	/			
										<u></u>	8.	· /			
12)	H	ask	4	yn (ch	on	٤								
phic	· r	1asl	n f	ne	tro	5	h	W) ap)5	bi	†- .	stri	ings	
•															
							7	•		ľ					_
eter	-Min	nist	me,	, p	اط	رعا	, '' <i>(</i>	4ne	om	, "	m	9 DN	۲۲	•	
0,3	1 }	7		->	ş	٥,	1	9							
F					_	₹		ر ال	st	てか	95	, f	ler	19t	h (
			. 11	ctr:	10.0	. 1									
											יי כי	,			
da	"m	255	age	di	ges	ť,	1	un	etre	'n,					
leng	yth:	5 4	re	d	=) a	8,	16	0,	2	56	5 ر	:12) Ł	>51
: Y_	A	140	nc	Ca	n	COM	nρι	te	. h	. 1	fo	m	its		
ptio	ກຸ	C	mp	outo	tio	n 1	3	ef			t	Sp	براه	- †î	M
									28			7	1		
												3	w	t	CR
5H	A-	25	6					à	56	•					-
5h	7 - 1A-	3						a a	4,2	54,	38'	1,5	12		—
	((Ke	2CC	ak)		Óط	t 2	01	2						
	eter O : Ve ptio MI SH	enoth the etermination of the second of the	eterminist O, 15 Anyo ption. Co MD4 MD5 SHA-1 SHA-25 SHA-3	eterministra eterministra 0,13 a "message lengths are y. Anyone ption. Comp MD4 MD5 SHA-1 SHA-256 SHA-3 SHA-3	eterministre, partial a "message distribution and dempths are despection. Computo MD4 MD5 SHA-1 SHA-256	eterministric, publication, all string, al	ength to a fixed-length to a fixed-length; a "message digest" lengths are d = 12 v. Anyone can comption. Computation MD4 MD5 SHA-1 SHA-256 SHA-3 SHA-3	enyth to a fixed-lengt eterministic, public, " 0,13 *	eterministre, public, "rana 0,13> 30,13 a "message digest" func lengths are d = 128,16 eterministre, public, "rana all strings (of an lengths are d = 128,16 eterministre, public, "rana all strings (of an lengths are d = 128,16 eterministre, public, "rana all strings (of an lengths are d = 128,16 eterministre, public, "rana All strings (of an lengths are d = 128,16 eterministre, public, "rana all strings (of an lengths are d = 128,16 eterministre, public, "rana all strings (of an lengths are d = 128,16 eterministre, public, "rana all strings (of an lengths are d = 128,16 eterministre, public, "rana all strings (of an lengths are d = 128,16 eterministre, public, "rana lengths are d = 128,16 eterministre, public, "rana eterministre, public, "ran	ength to a fixed-length our eterministic, public, "random 0,13 -> 30,13 Call strings (of any lengths are d = 128,160, Ex. Anyone can compute h ption. Computation is effice MDH MD5 SHA-1 SHA-256 SHA-512 SHA-3 SHA-3 Z24,2	eterministre, public, "random" 0,13 -> 30,13 all strings (at any len lengths are d = 128,160,2 ey. Anyone can compute h ption. Computation is efficien MD4 MD5 SHA-1 SHA-1 SHA-3 SHA-3 224,256	ength to a fixed-length output eterministic, public, "random" mand of the strings of any length a "message digest" function. lengths are d = 128, 160, 256 iv. Anyone can compute h frogetion. Computation is efficient all 128 MDH MDH IDB IDB SHA-1 SHA-3 SHA-3 Zay, asc, 38	ength to a fixed-length output in eterministic, public, "random" mann 0,13	ength to a fixed-length output in an eterministic, public, "random" manner O, 15 -> 30,13 Lall strings of any length 30 la "message digest" function. lengths are d = 128,160,256,512 Ey. Anyone can compute h from its ption. Computation is efficient (poly d) MD4 MD5 SHA-1 SHA-256 SHA-31 SHA-31 224,256,384,512	all strings of any length > 0) I a "message digest" function. lengths are d = 128, 160, 256, 512 b Ty. Anyone can compute h from its ption. Computation is efficient (poly-timestern) MDH MDS SHA-1 SHA-256 SHA-512 SHA-512 SHA-3 MARCH 229, 356, 387, 512

											DA	TE:			
ER:											PAC	GE: L	-8,8	3	
	Ide	a) F	المداء	Func	tion	•	Ra.	dom			. 10		Ro)	
	290	4) '	19.571	-	,,,,,			4511		Tere	15		70	-	
	• Th	eore	tical	mod	<u>el -</u>	not	ael	rievo	:ble	. in	pra	chr	و		
	0-	1. /	/		-1-	n \				-					
	Orac	1e (17	the	SKy										
	•	^e ((eive:	in	puts	X	L	ret	441	15	out	pul	- h	(x)	
					'										
		TOP	any	/ X	Ε,	20,	15	-		h()	(11	= 6	7	bit.	•
		On	in D	+	ΧΕ	80	2,13	* .							
			•												
		if	*	not	in	00	or;								
		•	flic	co	in c	t	me	s to	, d	ete	c Imi	ine	h	(x)	
		•	rec	ord	()	K, b) (×	((in	b	ook				
		els	e °	ret	Mri	2 '		whe	برو	()	ارکا	\ i	h	000	K.
						•						1			
	• Gi	ves	rar	ndom	\ an	SW	er	eve	7	1) u	ne,	bu	1	पडट	
	b	ook	to	160	rd ()(EV	zuci	an	SWE	'VS .	(0	h	2		
												' '			
	0	ete	ermi	ni3t	15 8	Co	1 5	sten	t.						
		1		2	3	CON									
		orad	4 5			1 6	ok								
			1	**	7	ŋ									
			4			} !		×				-			
			7	4	س		h(x	7	F	3-!	>				
				7	1/2(2										
			-	-	ا اد			_	\perp						

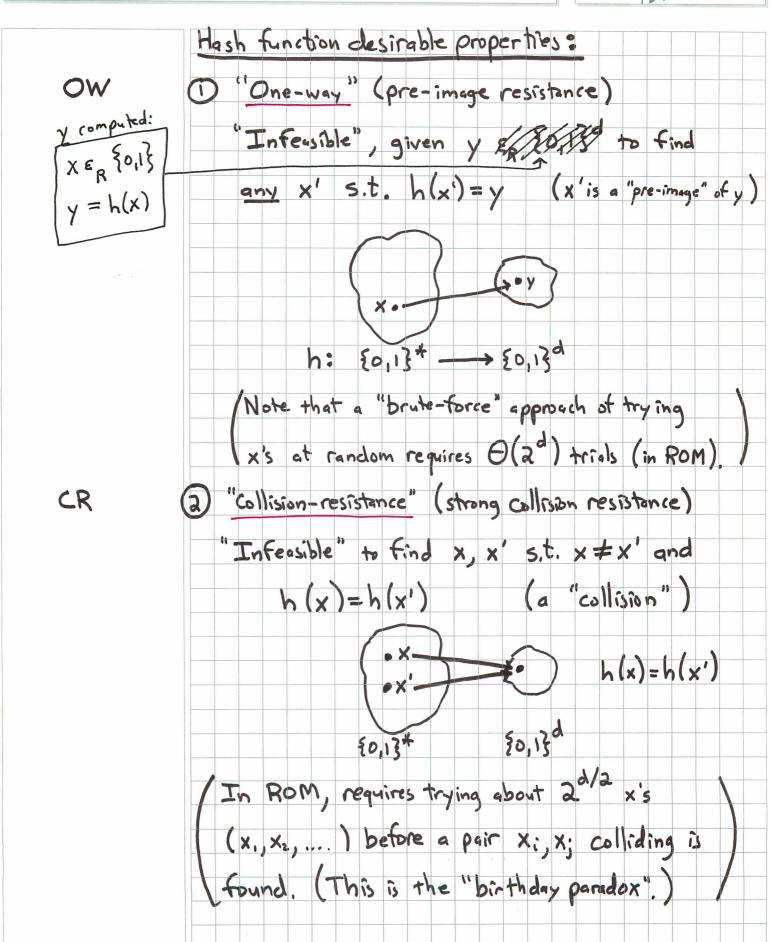
R:																PAG	E:	L.	8,	9		_
	1	۸4,		cr	e for	ava:	ohiz	<	he	me	5 6	276	0	M	vec	l s	ec	u٣	e i	n		
	1	1		1 1	- 1	- 1	- 1		1		1	i						Į.		l .		
	F	10	M	("	Rav	do	m C)sa	cle	M	sd	ય "),	W	بهر	به	45	Sul	nt	25		
		- v.	2			_	D A		-			0	^	0				. ,				
			5π	W CE	- 0		אט	•	1	ne	n	1	O	15	re	PIA	نده	1 6	Y			
		con	V21	tro	74	h	deo	6	inc	to	n	(e	٠, ٩	<	H	A-	. 2	50	,)	m		
			1						1			1	_									
		D C	cti	رو	, '	wh	izh	ذا	h	pe	fu	1	"f	१ऽ५	ud	<i>ه</i>	and	om	e	nou	ish'	1
			-		-					•												
		-																				
			-																			_
																						_
						-																
				-				-														
							-					_										
					_	-																
																						_
			-																			
																			-			
																						_
		-		-				-	-			-					-					_
			-			-			-												-	_
						+			-	-												-

TOPIC:

DATE:

FILE UNDER

PAGE: L9,20



TOPIC:	DATE:
FILE UNDER:	PAGE: L9, 3
	Note that collisions are unavoidable since
	[\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
	180,134 = 24
	Birthday paradox detail:
	If we hash x, x2,, xn (distinct strings)
	$E(\#collisions) = \sum_{i} P_{i}(h(x_{i}) = h(x_{i}))$
	$E(\#co) _{i \ge i \ne j} P_r(h(x_i) = h(x_j))$
	$= \binom{n}{2} \cdot 2^{-d} \qquad \text{[if h "uniform"]}$
	2 n ² . 2
	(4u))/2 1/2

This is > 1 when n > 2

for a 160-bit output.

dramitically reduced.

The birthday paradox is the reason why hash function

outputs are generally twice as big as you might naively

With some tricks, memory requirements can be

expect; you only get 80 bits of security (w.r.t. CR)

	DATE:
NDER:	PAGE: L9,4
TCR	(3) "Weak callision resistance" (target callision resistance 2nd pre-image resistance "Infersible", given $x \in \{0,1\}$ to find $x' \neq x$ s.t. $h(x) = h(x')$.
	Like CR, but one pre-image given & fixed. [In ROM, can find x' in time \(\theta(a^d))\) (as for OW, since knowing x doesn't help in ROM) to find x').
PRF	Pseudo-rundomness "In is indistinguishable under black-box access from a random pracle"
	To make this notion workable, really need a family of hash functions, one of which is chosen at random. A single, fixed, public hash function is easy to identify
NM	5) Non-malleability "Infeasible", given h(x), to produce
	h(x') where x and x' are "related" (e.g. $x'=x+1$).
	These are informal definitions

	DATE:
NDER:	PAGE: 19.5
	Theremon To by CP than by TCP
	Theorem: If h is CR, then h is TCR. (But Converse doesn't hold.)
	Theorem: h is OW () h is CR
	(neither implication holds) But if h "compresses", then CR => OW.
	Hash function applications
	D Password storage (For login)
	· Store h (PW), not PW, on computer
	 Store h (PW), not PW, on computer when user logs in, check hash of his PW agaist table. Disclosure of h (PW) should not reveal
	PW (or any equivalent pre-image)
	• Need OW
	(5) File mod firston detector
	(e.g. on off-line DVD) • Can check if F has been modified by
	(e.g. on off-line DVD)
	· Can check if F has been modified by
	recomputing h(F) • need WCR (aka TCR)
	(Adversary wants to change F but not h (F).)
	· Hashes of downloadable software = equivalent problem.
	hasnes of download tole software - equivalent problem.

TOPIC:	
10110.	

DATE:

FILE UNDER:

PAGE: L9.6

3	Digi	tal	sigr	ish	unes	('	ha	sh	& s	jjq	າ ")					
	PKA														424	ĩon)
	SK _A																
	Sign																M
	Veri																
	Adv									•							
•	For									9							
					(5											מא (7
	Veri																
	In																•
	Ne									•							
				-	h												
															.)		
	Don	`t /	n ec	d)
				-				3									,
														-			
,																	

TOPIC:	DATE:

FILE UNDER:

PAGE: LIIO

Dommitments Alice has value x (e.g. auction bid) Alice computes C(x) ("commitment to x") L submits C(x) as her "sealed bid" When bidding has closed, Alice should be able to "open" C(x) to reveal x Binding property: Alice should not be able to open C(x) in more than one way! (she is committed to just one x.) Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say How: C(x) = h(r x) r & 80,115
 Alice has value x (e.g. auction bid) Alice computes ((x) ('commitment to x") L submits C(x) as her "sealed bid" When bidding has closed, Alice should be able to 'open" C(x) to reveal x Binding property: Alice should not be able to open C(x) in more than one way! (she is committed to just one x.) Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say How: C(x) = h(r x) r = 80.15
Alice computes ((x) ("commitment to x") & submits ((x) as her "sealed bid" • When bidding has closed, Alice should be able +o "open" ((x) to reveal x • Binding property: Alice should not be able to open ((x) in more than one way! (she is committed to just one x.) • Secrecy (hiding): Auctioneer (or anyone else) seeing ((x) should not learn anything about x. • Non-malleability: Given ((x), it shouldn't be possible to produce ((x+1), say, • How:
Alice computes ((x) ("commitment to x") & submits ((x) as her "sealed bid" • When bidding has closed, Alice should be able +> "open" ((x) to reveal x • Binding property: Alice should not be able to open ((x) in more than one way! (she is committed to just one x.) • Secrecy (hiding): Auctioneer (or anyone else) seeing ((x) should not learn anything about x. • Non-malleability: Given ((x), it shouldn't be possible to produce ((x+1), say, • How: O(x) = h(r ((x)) r & for
Alice computes ((x) ("commitment to x") & submits ((x) as her "sealed bid" • When bidding has closed, Alice should be able +> "open" ((x) to reveal x • Binding property: Alice should not be able to open ((x) in more than one way! (she is committed to just one x.) • Secrecy (hiding): Auctioneer (or anyone else) seeing ((x) should not learn anything about x. • Non-malleability: Given ((x), it shouldn't be possible to produce ((x+1), say, • How: O(x) = h(r ((x)) r & for
A submits $C(x)$ as her "sealed bid" • When bidding has closed, Alie should be able +> "open" $C(x)$ to reveal x • Binding property: Alice should not be able to open $C(x)$ in more than one way! (she is committed to just one x .) • Secrecy (hiding): Auctioneer (or anyone else) seeing $C(x)$ should not learn anything about x . • Non-malleability: Given $C(x)$, it shouldn't be possible to produce $C(x+1)$, say • How: $C(x) = h(r(x))$ • $C(x) = h(r(x))$
• When bidding has closed, Alice should be able + b 'open" C(x) to reveal x • Binding property: Alice should not be able to open C(x) in more than one way! (She is committed to just one x.) • Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. • Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say • How: C(x) = h(r(x)) r = 50,13
Binding property: Alrice should not be able to open C(x) in more than one way! (she is committed to just one x.) Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say How: C(x) = h(r(x) r = 50.15
Binding property: Alice should not be able to open C(x) in more than one way! (she is committed to just one x.) Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say How: C(x) = h(r(x)) r = 50,15
open C(x) in more than one way! (she is committed to just one x.) • Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. • Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say • How: C(x) = h(r/(x) r & 30,15
open C(x) in more than one way! (she is committed to just one x.) • Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. • Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say • How: C(x) = h(r (x) r = \frac{3}{2}0.15
• Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. • Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say • How: C(x) = h(r x) r = {0,15}
Secrecy (hiding): Auctioneer (or anyone else) seeing C(x) should not learn anything about x. Non-malleability: Given C(x), it shouldn't be possible to produce C(x+1), say, How: C(x) = h(r x) r = {0,15}
seeing $C(x)$ should not learn anything about x . • Non-malleability: Given $C(x)$, it shouldn't be possible to produce $C(x+1)$, say, • How: $C(x) = h(r x) r \in \{0,1\}$
• Non-malleability: Given $C(x)$, it shouldn't be possible to produce $C(x+1)$, say, • How: $C(x) = h(r(x)) r \in \{0,1\}$
· Non-malleability: Given &(x), it shouldn't be possible to produce &(x+1), say, · How: C(x) = h(r(x) r = \frac{2}{2}, 1\frac{2}{3}
• How: $C(x) = h(r x) r \in \{0,1\}$
· How: C(x) = h(r//x) r = 50,13
$C(x) = h(r(x) r \in \{0,1\}$
$C(x) = h(r(x) r \in \{0,1\}$
$C(x) = h(r(x) r \in \{0,1\}$
The same cover to X X
To open: reveal r &x
0 81 - 10 - 14 - 14 - 17 - 17 - 17 - 17 - 17 - 17
· Note that this method is randomized (as it
must be for secrecy.
· Need: OW, CR, NM
(really need more, for secrecy, as C(x) should
not reveal partial information about x, even.)
110, lestal barlist lucion and x' socii.