CSE107: Intro to Modern Cryptography

https://cseweb.ucsd.edu/classes/sp22/cse107-a/

Emmanuel Thomé

May 19, 2022

Lecture 14b

Passwords and password-authenticated key exchange

Recap from Tuesday

Passwords and PAKE

Plan

Recap from Tuesday

Passwords and PAKE

Forward secrecy

Definition (Forward Secrecy)

Forward secrecy asks that exposure of $sk[B]$ does not allow recovery of session keys K exchanged prior to the time of exposure.

FS is achieved using the DH key exchange inside the session key exchange protocol.

Forward secrecy is considered necessary in modern session key exchange, and is present in the TLS 1.3 protocol.

Session-key exchange protocols using DH for forward secrecy are often called authenticated DH key exchange protocols.

Protocol KE3

Let $G = \langle g \rangle$ be a cyclic group of order m in which the CDH problem is hard.

$$\xrightarrow{A, g^a}$$

$$\xleftarrow{B, g^b, \mathrm{CERT}[B], \mathsf{Sig}_B(A \| B \| g^a \| g^b), \mathsf{MAC}_M(1 \| A \| B \| g^a \| g^b)}$$

$$\xrightarrow{\mathsf{MAC}_M(0 \| A \| B \| g^a \| g^b)}$$

Here $a, b \xleftarrow{\$} \mathbb{Z}_m$ are chosen by A, B, respectively, and g^a, g^b play the role of nonces.

$\mathsf{Sig}_B(X)$ is B's signature on X, computed under $sk[B]$ and verifiable under the $pk[B]$ that is in $\mathrm{CERT}[B]$.

Let $L = g^{ab}$ be the DH key. Then session key is $K = \mathbf{H}_1(A \| B \| g^a \| g^b \| L)$ and MAC key is $M = \mathbf{H}_2(A \| B \| g^a \| g^b \| L)$ where $\mathbf{H}_1, \mathbf{H}_2$ are as before.

Protocol KE3

$$\xrightarrow{A, g^a}$$

$$\xleftarrow{B, g^b, \mathrm{CERT}[B], \mathsf{Sig}_B(A\|B\|g^a\|g^b), \mathsf{MAC}_M(1\|A\|B\|g^a\|g^b)}$$

$$\xrightarrow{\mathsf{MAC}_M(0\|A\|B\|g^a\|g^b)}$$

There is no public-key encryption used here, only signatures.

Compromise of $sk[B]$ only gives E the ability to forge signatures. Even given $sk[B]$, it cannot recover the DH key $L = g^{ab}$ from a prior exchange, and thus cannot distinguish from random the session key $K = \mathbf{H}_1(A\|B\|g^a\|g^b\|L)$.

Accordingly this provides forward secrecy.

This is roughly the core of the unilateral session-key exchange in the TLS 1.3 handshake.

Plan

Recap from Tuesday

Passwords and PAKE

Passwords

A password is a human-memorizable key.

Attackers can form a set D of possible passwords called a dictionary such that

- If the target password pwd is in D, and also
- The attacker knows $\overline{\text{pwd}} = f(\text{pwd})$, the image of pwd under some public function f,

then the target password pwd can be found via:

For all $\text{pwd}' \in D$ do
    If $f(\text{pwd}') = \overline{\text{pwd}}$ then return pwd'

This is called a dictionary, or brute-force, attack.

Password usage

Passwords are in widespread use for client authentication to Internet services and servers like `gmail`, Amazon, Internet banking, ...

Most of us have more passwords than we can remember.

Passwords are communicated over TLS. The main threat is dictionary attacks arising from the adversary obtaining the image $\overline{\text{pwd}} = f(\text{pwd})$ of the target password pwd under some public function f.

Studies show that many users select poor passwords, meaning ones that fall into attacker dictionaries. And attackers get better and better at making dictionaries. So preventing dictionary attacks is important for security.

Popular passwords

In 2016, the 25 most common passwords made up more than 10% of surveyed passwords, with the most common making up 4%.

Top 25 most common passwords by year according to SplashData

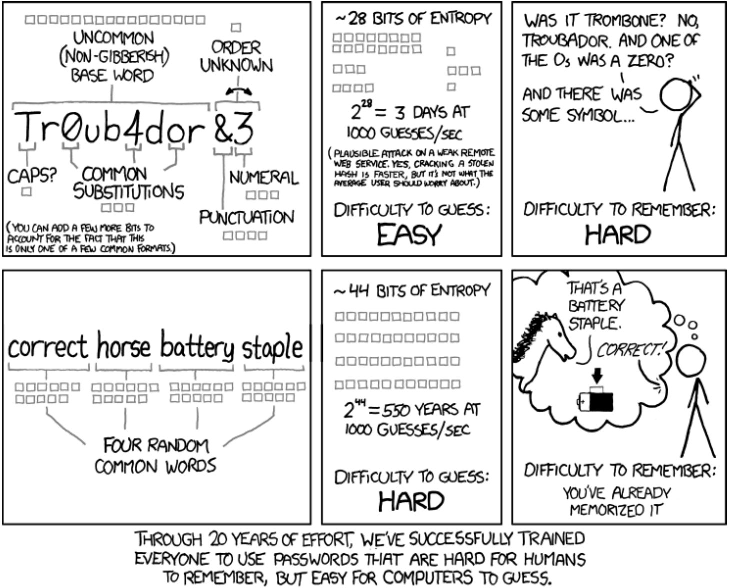
Rank	2011[4]	2012[5]	2013[6]	2014[7]	2015[8]	2016[3]	2017[9]	2018[10]
1	password	password	123456	123456	123456	123456	123456	123456
2	123456	123456	password	password	password	password	password	password
3	12345678	12345678	12345678	12345	12345678	12345	12345678	123456789
4	qwerty	abc123	qwerty	12345678	qwerty	12345678	qwerty	12345678
5	abc123	qwerty	abc123	qwerty	12345	football	12345	12345
6	monkey	monkey	123456789	123456789	123456789	qwerty	123456789	111111
7	1234567	letmein	111111	1234	football	1234567890	letmein	1234567
8	letmein	dragon	1234567	baseball	1234	1234567	1234567	sunshine
9	trustno1	111111	iloveyou	dragon	1234567	princess	football	qwerty
10	dragon	baseball	adobe123[a]	football	baseball	1234	iloveyou	iloveyou
11	baseball	iloveyou	123123	1234567	welcome	login	admin	princess
12	111111	trustno1	admin	monkey	1234567890	welcome	welcome	admin
13	iloveyou	1234567	1234567890	letmein	abc123	solo	monkey	welcome
14	master	sunshine	letmein	abc123	111111	abc123	login	666666
15	sunshine	master	photoshop[a]	111111	1qaz2wsx	admin	abc123	abc123
16	ashley	123123	1234	mustang	dragon	121212	starwars	football
17	bailey	welcome	monkey	access	master	flower	123123	123123
18	passw0rd	shadow	shadow	shadow	monkey	passw0rd	dragon	monkey
19	shadow	ashley	sunshine	master	letmein	dragon	passw0rd	654321
20	123123	football	12345	michael	login	sunshine	master	!@#$%^&*
21	654321	jesus	password1	superman	princess	master	hello	charlie
22	superman	michael	princess	696969	qwertyuiop	hottie	freedom	aa123456
23	qazwsx	ninja	azerty	123123	solo	loveme	whatever	donald
24	michael	mustang	trustno1	batman	passw0rd	zaq1zaq1	qazwsx	password1
25	Football	password1	000000	trustno1	starwars	password1	trustno1	qwerty123

Commonly heard gibberish

Over the years, recommendations about “password strength” have become ubiquitous.

- “your password must include uppercase and lowercase letters, digits, two punctuation symbols”, etc.
- and “you must change your password every 12 months”.

XKCD 936



UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS:
EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER:
HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS:
HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER:
YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

NIST on passwords

NIST SP800-63B, revised in 2020 (§5.1.1.2 Memorized Secret Verifiers).

- *Verifiers SHOULD NOT impose other composition rules (e.g., requiring mixtures of different character types or prohibiting consecutively repeated characters) for memorized secrets.*
- *Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically). However, verifiers SHALL force a change if there is evidence of compromise of the authenticator.*
- *Verifiers SHOULD permit claimants to use "paste" functionality when entering a memorized secret. This facilitates the use of password managers, which are widely used and in many cases increase the likelihood that users will choose stronger memorized secrets.*

Password managers: lastpass, keepass, bitwarden, pass, ...

PAKE

A protocol for Password Authenticated Key Exchange (PAKE) assumes client A has a password pwd and server B has either pwd or its hash under a public hash function.

The parties interact to arrive at a common session key K satisfying authenticity, secrecy, forward secrecy and also *security against off-line dictionary attacks*.

This means the protocol never reveals an image $\overline{\mathrm{pwd}} = f(\mathrm{pwd})$ of pwd under a public function f. So even if the password is in the dictionary, the off-line dictionary attack is infeasible.

Roughly, one adversary interaction with one of the parties can eliminate at most one candidate password from the dictionary.

Authentication here is mutual, and no PKI / certificates are assumed.

Protocol KE4

$$\xrightarrow{A,g^a}$$
$$\xleftarrow{B,g^b,\mathrm{MAC}_M(1\|A\|B\|g^a\|g^b)}$$
$$\xrightarrow{\mathrm{MAC}_M(0\|A\|B\|g^a\|g^b)}$$

Client A has password pwd that is known to server B.

Let $L = g^{ab}$ be the DH key. Then the session key and MAC keys are $K = \mathbf{H}_1(A\|B\|g^a\|g^b\|L\|\mathrm{pwd})$ and $M = \mathbf{H}_2(A\|B\|g^a\|g^b\|L\|\mathrm{pwd})$, respectively.

Is this secure against dictionary attack?

Protocol KE4

$$\xrightarrow{A, g^a}$$

$$\xleftarrow{B, g^b, \mathrm{MAC}_M(1\|A\|B\|g^a\|g^b)}$$

$$\xrightarrow{\mathrm{MAC}_M(0\|A\|B\|g^a\|g^b)}$$

A successful dictionary attack by adversary E is possible, as follows:

$$\xrightarrow{A, g^a}$$

$$\xleftarrow{B, g^b, \mathrm{MAC}_M(1\|A\|B\|g^a\|g^b)}$$

E has A, B, g^a, g^b and also $L = g^{ab} = (g^b)^a$. Let

$$f(\mathrm{pwd}) = \mathrm{MAC}_{\mathbf{H}_2(A\|B\|g^a\|g^b\|L\|\mathrm{pwd})}(A\|B\|g^a\|g^b)\ .$$

This f is a public function of the password, allowing E to mount the dictionary attack.

History and status of PAKE

The first protocols were by Bellovin and Merrit, 1992.

Definitions and proven-secure protocols begin with [BPR00].

Large literature.

A representative modern PAKE protocol is OPAQUE [JKX18].

CSE107: Intro to Modern Cryptography

https://cseweb.ucsd.edu/classes/sp22/cse107-a/

Emmanuel Thomé

May 19, 2022

Lecture 15a

Advanced primitives and protocols

Commitment schemes

Homomorphic encryption

Advanced primitives and protocols

A large body of work on cryptography for goals beyond secure communication.

Usually concerned with privacy in broader settings.

Encompasses computing on encrypted data, secure two- and multi- party computation protocols, zero-knowledge ...

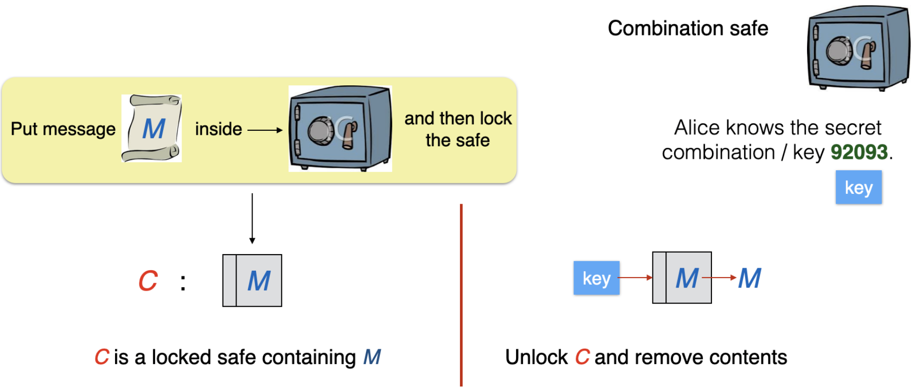
We start with safes and commitment schemes ...

Plan

Commitment schemes

Homomorphic encryption

Safes and their properties



Hiding: Without key K, one cannot recover the content of locked safe C.

Binding: A single, locked safe C cannot admit two keys K_1, K_2 that open it to reveal different content M_1, M_2.

Commitment schemes

Commitment schemes are, at first cut, a cryptographic (mathematical, digital, ...) way to realize safes.

To be more accurate, a commitment scheme is a cryptographic primitive whose definition formalizes requirements called hiding and binding. A safe is a rough physical analogy, or metaphor, for a commitment scheme.

As with all metaphors, it has its limits, so try to understand commitment schemes via the definitions rather than solely via the metaphor.

Zen saying: The finger pointing at the moon is not the moon ...

Commitment schemes are used in many protocols, including zero-knowledge protocols.

Syntax of a Commitment Scheme

A commitment scheme $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ is a triple of algorithms that operate as follows:

- $\pi \xleftarrow{\$} \mathcal{P}$ — a trusted party runs the parameter generation algorithm $\mathcal{P}$ to generate public parameters π
- $(K, C) \xleftarrow{\$} \mathcal{C}_\pi(M)$ — apply commitment algorithm $\mathcal{C}$ to message M to obtain a commitment C to M along with a decommitment (or opening) key K.
- $d \leftarrow \mathcal{V}_\pi(C, M, K)$ — apply verification algorithm $\mathcal{V}$ to commitment C, candidate message M and key K to obtain a decision $d \in \{0, 1\}$ as to whether C is a commitment to M.

The correctness requirement is that, for all π that may be output by $\mathcal{P}$, and all messages M from the underlying message space, we have $d = 1$ with probability 1 when $(K, C) \xleftarrow{\$} \mathcal{C}_\pi(M)$ and $d \leftarrow \mathcal{V}_\pi(C, M, K)$.

Hiding security

Let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be a commitment scheme.

Game $\mathrm{HIDE}_{\mathcal{CS}}$

procedure Initialize
$\pi \xleftarrow{\$} \mathcal{P}$; $b \xleftarrow{\$} \{0,1\}$
return π

procedure LR(M_0, M_1)
$(K, C) \xleftarrow{\$} \mathcal{C}_\pi(M_b)$
return C

procedure Finalize(b')
return $(b = b')$

Definition (hiding-advantage)

The hiding-advantage of an adversary A is

$$\mathbf{Adv}^{\mathrm{HIDE}}_{\mathcal{CS}}(A) = 2 \cdot \Pr\left[\mathrm{HIDE}^{A}_{\mathcal{CS}} \Rightarrow \mathsf{true}\right] - 1 .$$

Hiding security asks that an adversary having C but not K should not learn even partial information about the message M.

Binding security

Let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be a commitment scheme.

Game $\mathrm{BIND}_{\mathcal{CS}}$

procedure Initialize
$\pi \xleftarrow{\$} \mathcal{P}$
return π

procedure Finalize(C, M_0, M_1, K_0, K_1)
$v_0 \leftarrow \mathcal{V}_\pi(C, M_0, K_0)$
$v_1 \leftarrow \mathcal{V}_\pi(C, M_1, K_1)$
return $((v_0 = 1)$ and $(v_1 = 1)$ and $(M_0 \neq M_1))$

Definition (binding-advantage)

The binding-advantage of an adversary A is

$$\mathbf{Adv}^{\mathrm{BIND}}_{\mathcal{CS}}(A) = \Pr\left[\mathrm{BIND}^{A}_{\mathcal{CS}} \Rightarrow \mathsf{true}\right] .$$

Binding security asks that an adversary be unable to create a commitment C that it can open to two different messages.

Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_\pi(M)$	**Alg** $\mathcal{V}_\pi(C, M, K)$
$\pi \leftarrow \varepsilon$	$K \xleftarrow{\$} \mathcal{K}$; $C \xleftarrow{\$} \mathcal{E}_K(M)$	if $\mathcal{D}_K(C) = M$ then return 1
return π	return (K, C)	else return 0

Q: Is this hiding?

Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

<u>**Alg** $\mathcal{P}$</u>	<u>**Alg** $\mathcal{C}_\pi(M)$</u>	<u>**Alg** $\mathcal{V}_\pi(C, M, K)$</u>
$\pi \leftarrow \varepsilon$	$K \xleftarrow{\$} \mathcal{K}$; $C \xleftarrow{\$} \mathcal{E}_K(M)$	if $\mathcal{D}_K(C) = M$ then return 1
return π	return (K, C)	else return 0

Q: Is this hiding?

YES, since $\mathcal{SE}$ is IND-CPA.

Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

<u>**Alg** $\mathcal{P}$</u>	<u>**Alg** $\mathcal{C}_\pi(M)$</u>	<u>**Alg** $\mathcal{V}_\pi(C, M, K)$</u>
$\pi \leftarrow \varepsilon$	$K \xleftarrow{\$} \mathcal{K}$; $C \xleftarrow{\$} \mathcal{E}_K(M)$	if $\mathcal{D}_K(C) = M$ then return 1
return π	return (K, C)	else return 0

Q: Is this binding?

Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_\pi(M)$	**Alg** $\mathcal{V}_\pi(C, M, K)$
$\pi \leftarrow \varepsilon$	$K \xleftarrow{\$} \mathcal{K}$; $C \xleftarrow{\$} \mathcal{E}_K(M)$	if $\mathcal{D}_K(C) = M$ then return 1
return π	return (K, C)	else return 0

Q: Is this binding?

Not necessarily. For schemes like CTR\$ or CBC\$, the following adversary will have high binding advantage:

adversary $A(\pi)$

$K_0, K_1 \xleftarrow{\$} \{0,1\}^k$; $M_0 \xleftarrow{\$} \{0,1\}^L$; $C \xleftarrow{\$} \mathcal{E}_{K_0}(M_0)$; $M_1 \leftarrow \mathcal{D}_{K_1}(C)$

return (C, M_0, M_1, K_0, K_1)

Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_\pi(M)$	**Alg** $\mathcal{V}_\pi(C, M, K)$
$\pi \leftarrow \varepsilon$	$K \xleftarrow{\$} \mathcal{K}$; $C \xleftarrow{\$} \mathcal{E}_K(M)$	if $\mathcal{D}_K(C) = M$ then return 1
return π	return (K, C)	else return 0

Q: Is this binding if we additionally assume $\mathcal{SE}$ is INT-CTXT-secure?

Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_\pi(M)$	**Alg** $\mathcal{V}_\pi(C, M, K)$
$\pi \leftarrow \varepsilon$	$K \xleftarrow{\$} \mathcal{K}$; $C \xleftarrow{\$} \mathcal{E}_K(M)$	if $\mathcal{D}_K(C) = M$ then return 1
return π	return (K, C)	else return 0

Q: Is this binding if we additionally assume $\mathcal{SE}$ is INT-CTXT-secure?

The above attack may no longer work. But the answer to the above question is NO.

If $\mathcal{SE}$ is *robust* [ABN10,FLPQ13,FOR17] or *committing* [GLR17] then $\mathcal{CS}$ will be binding.

Commitment from hashing

Let $\mathbf{H}\colon \{0,1\}^* \to \{0,1\}^\ell$ be a collision-resistant hash function and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_\pi^{\mathbf{H}}(M)$	**Alg** $\mathcal{V}_\pi^{\mathbf{H}}(C, M, K)$
$\pi \leftarrow \varepsilon$	$C \leftarrow \mathbf{H}(M)$	If $((C = \mathbf{H}(M))$ and $(M = K))$
return π	$K \leftarrow M$	then return 1
	return (K, C)	Else return 0

Q: Is this binding?

Commitment from hashing

Let $\mathbf{H}\colon \{0,1\}^* \to \{0,1\}^\ell$ be a collision-resistant hash function and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_\pi^{\mathbf{H}}(M)$	**Alg** $\mathcal{V}_\pi^{\mathbf{H}}(C, M, K)$
$\pi \leftarrow \varepsilon$	$C \leftarrow \mathbf{H}(M)$	If $((C = \mathbf{H}(M))$ and $(M = K))$
return π	$K \leftarrow M$	then return 1
	return (K, C)	Else return 0

Q: Is this binding?

YES, since **H** is collision resistant.

Commitment from hashing

Let $\mathbf{H}$: $\{0,1\}^* \to \{0,1\}^\ell$ be a collision-resistant hash function and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

<u>**Alg** $\mathcal{P}$</u>	<u>**Alg** $\mathcal{C}_\pi^{\mathbf{H}}(M)$</u>	<u>**Alg** $\mathcal{V}_\pi^{\mathbf{H}}(C, M, K)$</u>
$\pi \leftarrow \varepsilon$	$C \leftarrow \mathbf{H}(M)$	If $((C = \mathbf{H}(M))$ and $(M = K))$
return π	$K \leftarrow M$	then return 1
	return (K, C)	Else return 0

Q: Is this hiding?

Commitment from hashing

Let $\mathbf{H}\colon \{0,1\}^* \to \{0,1\}^\ell$ be a collision-resistant hash function and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

<u>**Alg** $\mathcal{P}$</u>	<u>**Alg** $\mathcal{C}_\pi^{\mathbf{H}}(M)$</u>	<u>**Alg** $\mathcal{V}_\pi^{\mathbf{H}}(C, M, K)$</u>
$\pi \leftarrow \varepsilon$	$C \leftarrow \mathbf{H}(M)$	If $((C = \mathbf{H}(M))$ and $(M = K))$
return π	$K \leftarrow M$	then return 1
	return (K, C)	Else return 0

Q: Is this hiding?

NO, since $\mathcal{C}$ is deterministic. Specifically, the following adversary A has $\mathbf{Adv}^{\mathrm{HIDE}}_{\mathcal{CS}}(A) = 1$:

adversary $A(\pi)$
$C_1 \leftarrow \mathbf{LR}(0, 1)$; $C_2 \leftarrow \mathbf{LR}(1, 1)$
If $(C_1 = C_2)$ then return 1 else return 0

Commitment from hashing

Let $\mathbf{H}$: $\{0,1\}^* \to \{0,1\}^\ell$ and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

<u>**Alg** $\mathcal{P}$</u>	<u>**Alg** $\mathcal{C}^{\mathbf{H}}_\pi(M)$</u>	<u>**Alg** $\mathcal{V}^{\mathbf{H}}_\pi(C, M, K)$</u>
$\pi \leftarrow \varepsilon$	$K \xleftarrow{\$} \{0,1\}^\ell$	If $(C = \mathbf{H}(K \| M))$
return π	$C \leftarrow \mathbf{H}(K \| M)$	then return 1
	return (K, C)	Else return 0

This is binding if $\mathbf{H}$ is collision-resistant (CR). Note: ℓ must be fixed!

One can give an example of CR $\mathbf{H}$ such that it is not hiding. But for "real" H such as SHA256 it seems to be hiding in the sense that no attacks are known.

Commitment from DL

Let $G = \langle g \rangle$ be a cyclic group whose order m is prime. Let $\mathbf{H}$: $\{0,1\}^* \to \mathbb{Z}_m$ and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_h^{\mathbf{H}}(M)$	**Alg** $\mathcal{V}_h^{\mathbf{H}}(C, M, K)$
$x \xleftarrow{\$} \mathbb{Z}_m$	$K \xleftarrow{\$} \mathbb{Z}_m$	If $(C = g^{\mathbf{H}(M)} h^K)$ then return 1
$h \leftarrow g^x$	$C \leftarrow g^{\mathbf{H}(M)} h^K$	Else return 0
$\mathrm{return}\ h$	$\mathrm{return}\ (K, C)$	

This is binding if DL is hard in G and $\mathbf{H}$ is collision-resistant (CR).

This is unconditionally hiding, meaning $\mathbf{Adv}_{\mathcal{CS}}^{\mathrm{HIDE}}(A) = 0$ for all A.

Commitment from DL

Let $G = \langle g \rangle$ be a cyclic group whose order m is prime. Let $\mathbf{H}\colon \{0,1\}^* \to \mathbb{Z}_m$ and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_h^{\mathbf{H}}(M)$	**Alg** $\mathcal{V}_h^{\mathbf{H}}(C, M, K)$
$x \xleftarrow{\$} \mathbb{Z}_m$	$K \xleftarrow{\$} \mathbb{Z}_m$	If $(C = g^{\mathbf{H}(M)} h^K)$ then return 1
$h \leftarrow g^x$	$C \leftarrow g^{\mathbf{H}(M)} h^K$	Else return 0
return h	return (K, C)	

The Pedersen commitment scheme [Pe91] is the special case where the message space is $\mathbb{Z}_m$ and $\mathbf{H}(M) = M$.

The Pedersen scheme is *homomorphic:* If $C_1 = g^{M_1} h^{K_1}$ is a commitment to M_1 and $C_2 = g^{M_2} h^{K_2}$ is a commitment to M_2 then $C_1 C_2 = g^M h^K$ is a commitment to $M = (M_1 + M_2) \bmod m$, with $K = (K_1 + K_2) \bmod m$.

Commitment from DL

Let $G = \langle g \rangle$ be a cyclic group whose order m is prime. Let $\mathbf{H}$: $\{0,1\}^* \to \mathbb{Z}_m$ and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

Alg $\mathcal{P}$	**Alg** $\mathcal{C}_h^{\mathbf{H}}(M)$	**Alg** $\mathcal{V}_h^{\mathbf{H}}(C, M, K)$
$x \xleftarrow{\$} \mathbb{Z}_m$	$K \xleftarrow{\$} \mathbb{Z}_m$	If $(C = g^{\mathbf{H}(M)} h^K)$ then return 1
$h \leftarrow g^x$	$C \leftarrow g^{\mathbf{H}(M)} h^K$	Else return 0
return h	return (K, C)	

The Pedersen commitment scheme [Pe91] is the special case where the message space is $\mathbb{Z}_m$ and $\mathbf{H}(M) = M$.

The Pedersen scheme is *homomorphic:* If $C_1 = g^{M_1} h^{K_1}$ is a commitment to M_1 and $C_2 = g^{M_2} h^{K_2}$ is a commitment to M_2 then $C_1 C_2 = g^M h^K$ is a commitment to $M = (M_1 + M_2) \bmod m$, with $K = (K_1 + K_2) \bmod m$.

What is x good for?

Commitment from DL

Let $G = \langle g \rangle$ be a cyclic group whose order m is prime. Let $\mathbf{H}$: $\{0,1\}^* \to \mathbb{Z}_m$ and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

<u>**Alg** $\mathcal{P}$</u>	<u>**Alg** $\mathcal{C}_h^{\mathbf{H}}(M)$</u>	<u>**Alg** $\mathcal{V}_h^{\mathbf{H}}(C, M, K)$</u>
$x \xleftarrow{\$} \mathbb{Z}_m$	$K \xleftarrow{\$} \mathbb{Z}_m$	If $(C = g^{\mathbf{H}(M)} h^K)$ then return 1
$h \leftarrow g^x$	$C \leftarrow g^{\mathbf{H}(M)} h^K$	Else return 0
return h	return (K, C)	

What is x good for?

Nobody knows x. A party that does know x can easily forge commitments (win the binding game).

Exercise: given two messages M and M', show that an adversary that knows x can compute K and K' such that C is a commitment to both (K, M) and (K', M').

Flipping a common coin

Alice and Bob are getting divorced. They want to flip a common, fair coin c whose outcome decides which of them keeps the waffle maker.

The naive protocol is for A to flip the coin c and send it to B:

$$\begin{array}{ccc} A & & B \\ c \xleftarrow{\$} \{0,1\} & \xrightarrow{\qquad c \qquad} & \end{array}$$

But this allows A to dictate the outcome. Unsurprisingly, she gets the waffle maker.

Flipping a common coin

Let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be a commitment scheme and consider the following protocol to flip a common coin c:

$$
\begin{array}{lcl}
\quad\quad A & & \quad\quad B \\
a \xleftarrow{\$} \{0,1\} & & \\
(K, C) \xleftarrow{\$} \mathcal{C}_\pi(a) & \xrightarrow{\quad C \quad} & b \xleftarrow{\$} \{0,1\} \\
 & \xleftarrow{\quad b \quad} & \\
 & \xrightarrow{\quad a,K \quad} & \text{if } \mathcal{V}_\pi(C, a, K) = 1 \text{ then } c \leftarrow a \oplus b \\
c \leftarrow a \oplus b & & \text{else } c \leftarrow \perp
\end{array}
$$

The hiding security of $\mathcal{CS}$ means that B cannot dictate the outcome c.

The binding security of $\mathcal{CS}$ means that A cannot dictate the outcome c.

Plan

Commitment schemes

Homomorphic encryption

Homomorphic encryption

Let $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, either symmetric or asymmetric.

We write ek for the encryption key and dk for the decryption key. In the symmetric case, they are the same.

We define the *homomorphic evaluation key* hk to be ek in the asymmetric case and ε in the symmetric case.

Let FC be a set (class) of functions. We write $\langle f \rangle$ for a description, for example as a circuit, of a function $f \in \mathrm{FC}$.

Homomorphic encryption

$\mathcal{HE}$ is a *homomorphic evaluation algorithm* for $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and FC if for all functions $f \in \mathrm{FC}$ and all messages $M_1, \ldots, M_m$, where m is the number of inputs of f, the following returns true with probability 1:

For $i = 1, \ldots, m$ do $C_i \stackrel{\$}{\leftarrow} \mathcal{E}_{ek}(M_i)$
$C \stackrel{\$}{\leftarrow} \mathcal{HE}_{hk}(\langle f \rangle, C_1, \ldots, C_m)$; $M \leftarrow \mathcal{D}_{dk}(C)$
Return $(M = f(M_1, \ldots, M_m))$

That is, C is an encryption of $f(M_1, \ldots, M_m)$.

Encryption scheme $\mathcal{ES}$ is homomorphic for the class of functions FC if there is an efficient homomorphic evaluation algorithm $\mathcal{HE}$ as above.

A fully homomorphic encryption (FHE) scheme is one that is homomorphic for the class FC of all functions.

Homomorphic encryption

Q: Isn't homomorphic evaluation always possible, via

Alg $\mathcal{HE}_{hk}(\langle f\rangle, C_1, \ldots, C_m)$

For $i = 1, \ldots, m$ do $M_i \leftarrow \mathcal{D}_{dk}(C_i)$

$M \leftarrow f(M_1, \ldots, M_m)$; $C \xleftarrow{\$} \mathcal{E}_{ek}(M)$; Return C

A: $\mathcal{HE}$ is not given dk. And the requirement that $\mathcal{HE}$ is efficient means that it is infeasible for it to compute dk from hk.

Security of homomorphic encryption

The primary security requirement for a homomorphic encryption scheme $\mathcal{ES}$ is simply IND-CPA.

Sometimes one wants the scheme to be function hiding (FH), which means that, on seeing $C \xleftarrow{\$} \mathcal{HE}_{hk}(\langle f \rangle, \mathcal{E}_{ek}(M_1), \ldots, \mathcal{E}_{ek}(M_m))$, one does not learn f. A game-based definition follows.

Sometimes one wants that homomorphically evaluated ciphertexts are distributed just like real ones, meaning the following are indistinguishable:

- $C \xleftarrow{\$} \mathcal{HE}_{hk}(\langle f \rangle, \mathcal{E}_{ek}(M_1), \ldots, \mathcal{E}_{ek}(M_m))$
- $C' \xleftarrow{\$} \mathcal{E}_{ek}(f(M_1, \ldots, M_m))$.

Extended key generation

Let $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, either symmetric or asymmetric. We define its *extended key-generation algorithm* $\overline{\mathcal{K}}$ via:

If $\mathcal{ES}$ is symmetric:	If $\mathcal{ES}$ is asymmetric:
Alg $\overline{\mathcal{K}}$	**Alg** $\overline{\mathcal{K}}$
$K \xleftarrow{\$} \mathcal{K}$	$(ek, dk) \xleftarrow{\$} \mathcal{K}$
$ek \leftarrow K$; $dk \leftarrow K$; $hk \leftarrow \varepsilon$	$hk \leftarrow ek$
Return (ek, dk, hk)	Return (ek, dk, hk)

This yields a unified syntax for symmetric and asymmetric schemes.

Function hiding security

Let $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme with homomorphic evaluation algorithm $\mathcal{HE}$ for the class of functions FC. Let A be an adversary.

Game $\mathrm{FH}_{\mathcal{ES},\mathcal{HE}}$

procedure Initialize
$b \xleftarrow{\$} \{0,1\}$; $i \leftarrow 0$;
$(ek, dk, hk) \xleftarrow{\$} \overline{\mathcal{K}}$;
Return hk

procedure Finalize(b')
return $(b = b')$

procedure Enc(M)
$i \leftarrow i + 1$; $M_i \leftarrow M$;
$C_i \xleftarrow{\$} \mathcal{E}_{ek}(M)$;
Return C_i

procedure LR$(i_1, \ldots, i_m, f_0, f_1)$
$C \xleftarrow{\$} \mathcal{HE}_{hk}(\langle f_b \rangle, C_{1_1}, \ldots, C_{i_m})$;
Return C

Function hiding security

In game $\mathrm{FH}_{\mathcal{ES},\mathcal{HE}}$, any **LR** query $i_1, \ldots, i_m, f_0, f_1$ must satisfy the following conditions:

- $f_0, f_1 \in \mathrm{FC}$
- m is the number of inputs of both f_0 and f_1
- $1 \le i_1, \ldots, i_m \le i$
- $|f_0(M_{1_1}, \ldots, M_{i_m})| = |f_1(M_{1_1}, \ldots, M_{i_m})|$.

Definition (fh-advantage)

The fh-advantage of A is

$$\mathbf{Adv}^{\mathrm{fh}}_{\mathcal{ES},\mathcal{HE}}(A) = 2 \cdot \Pr\left[\mathrm{FH}^{A}_{\mathcal{ES},\mathcal{HE}} \Rightarrow \mathsf{true}\right] - 1\ .$$

We (informally) say that $(\mathcal{ES}, \mathcal{HE})$ is FH-secure for FC if, as usual, any practical adversary A has low fh-advantage.

Homomorphic encryption can't be IND-CCA

If an encryption scheme $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is homomorphic for a non-trivial class of functions FC then it cannot be IND-CCA-secure.

Why? Assume the adversary A can find M_0, M_1 and $f \in \mathrm{FC}$ such that

1. $f(M_0) \neq M_0$ and $f(M_1) \neq M_1$
2. $f(M_0) \neq f(M_1)$

Then it can achieve $\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{ES}}(A) = 1$ via:

adversary $A(hk)$ // $hk = ek$ (asymmetric) or $hk = \varepsilon$ (symmetric)
$C \xleftarrow{\$} \mathbf{LR}(M_0, M_1)$; $C' \xleftarrow{\$} \mathcal{HE}_{hk}(\langle f \rangle, C)$; $M' \leftarrow \mathbf{Dec}(C')$
If $(M' = f(M_1))$ then return 1 else return 0

Condition (1) ensures $C' \neq C$ so the **Dec**-query is valid. Then (2) ensures that A's output is correct.

Possible usage of homomorphic encryption

Homomorphic encryption allows computing on encrypted data.

A picks keys ek, dk, hk, encrypts her data $M_1, \ldots, M_m$ under ek to get $C_1, \ldots, C_m$.

A uploads the ciphertexts and hk to in-the-cloud server B.

Later A can send $\langle f \rangle$ to B, who computes and returns $C \stackrel{\$}{\leftarrow} \mathcal{HE}_{hk}(\langle f \rangle, C_1, \ldots, C_m)$.

A now recovers $M = f(M_1, \ldots, M_m) \leftarrow \mathcal{D}_{dk}(C)$.