LIS-3353

Other Milkshakey Topics...

But...the NSA?

- All known NSA attacks/backdoors aren't against the actual method itself...

...but its implementation.

Random Numbers:

We need primes, but obviously we can't do them in order.

We need RANDOM NUMBERS.

but, can computers do random numbers?

Sure, you could bang on a calculator, but...

Good RNG's..

Pull from a number of different places..

- typing
- cameras

..etc.

..bad ones, aren't really random.

“Random number generator? Sure, why not try OURS? Wink wink.

So, what's using this now? A lot, with different levels of “reliability”

- SSL
- Bitlocker
- Whatsapp
- Signal

But the most interesting is:

Truecrypt!

- You can use it now.

- Makes you wiggle your mouse. Why?

- The FBI tried/tries to force people to give up their passwords. This is an interesting legal question. Is a password..

    - a “thing” like a key (then it can be forced)

    - Or “words?” (plead the ____th amendment)

    (also, what does this tell us about truecrypt)

But then, suddenly…

“WARNING: Using TrueCrypt is not secure as it may contain unfixed security issues”

“So use this different Windows one. Don't use the linux one at all. “

-hmmmm

Sign in a library:

“So far, we have NOT received any requests from the FBI to turn over your data..”

But then, one day, the sign is gone...

Also, passwords

- When you lose your password, what does the website do?

  Make you change it?

  Or

  Send you a copy of it.

If they're nice enough to send you your password, you can rest assured..

If they're nice enough to send you your password, you can rest assured..

that they suck at security:

Passwords

- If they send you a copy...they suck at security.

  Good websites CANT send you your password.

- Because they don't actually know it.

Obviously, if you're storing passwords, you want them "encrypted"

One way to do this:

1) Get their password

2) Save it on your computer

3) Then, encrypt it for safety.

...but wait

What do we REALLY need?

The actual password

or

simply:

Proof that they typed in the same thing both times?

What do we REALLY need?

Remember: “encrypting” something always yields:

UNIQUE GIBBERISH

“MyPassword123” >
ab18db351a3ed3849cca9839d98381ee63
92eeb391baa39d766290082812d9eceab

So, let's just switch it:

ANOTHER way to do this:

1) Get their password

2) ENCRYPT IT FIRST for safety.
3) Then save the ENCRYPTED password.
→
next time they log in?

So, let's just switch it:

next time they log in?

1) Get password
2) Encrypt it the same way, then compare the gibberish!

~~"MyPassword123" = "MyPassword123"~~

but instead..

So, let's just switch it:

"ab18db351a3ed3849cca9839d98381ee639
2eeb391baa39d766290082812d9eceab"

=

"ab18db351a3ed3849cca9839d98381ee639
2eeb391baa39d766290082812d9eceab"?

And "MyPassword123" IS NOT ON THE SERVER

But wait: Let's do one more thing?
Do we really need all of this?

"ab18db351a3ed3849cca9839d98381ee639
2eeb391baa39d766290082812d9eceab"

=

"ab18db351a3ed3849cca9839d98381ee639
2eeb391baa39d766290082812d9eceab"?

And "MyPassword123" IS NOT ON THE SERVER

But wait: Let's do one more thing? Do we really need all of this?

"ab18db351a3" = "ab18db351a3"*

As long as

- we use ALL the data in the original to get this number
- And it's STILL mathematically unliklely that two different passwords will yield the same short gibberish, we're good to go.

*you don't quite just cut a chunk off, but it's like this

But wait: Let's do one more thing? Do we really need all of this?

"ab18db351a3" = "ab18db351a3"*

ADVANTAGES:

- It's shorter
- Now you LITERALLY CANNOT "decrypt" it because you're missing some information. This is good!
- And, now – we can use this verification method on things other than passwords as well.

To Illustrate, first something slightly dumber...

File Verification

Presumption: The network (or person) is imperfect. The bytes we receive may not always be the exact ones that were sent.

Also: The network or verification is “slow”

We need a shorter, but verifiable, version of the data.

Basic Checksumming

The grocery list:

Cheese

- Crackers
- Eggs
- Ham
- Koala
- Mangoes
- Salt
- Underwear

Send the following...

- Cheese
- Ham
- Eggs
- Crackers
- Koala
- Salt
- Underwear
- Mangoes
- CHECKSUM.45

CHECKSUM.45 = CHECKSUM.45

If the reciever gets

- Cheese
- Ham
- Eggs
- Crackers
- Salt
- Underwear
- Mangoes
- CHECKSUM.45

CHECSUM.40 = CHECKSUM.45? *NO, SEND AGAIN.*

Hashing

Error checking/Checksumming.

One tiny change in the original still means BIG changes in the gibberish.

(MD5, which is fast, but not super-secure) is good for this)

Hashing

“Used to map data of arbitrary (big) size,

to data of fixed (small) size.”

Verification:

Hashing Uses

- Error Checking/Checksums
- Password “Storage”
- Bitcoin

Passwords

They don't store your password (your secret ingredient)

They just store the entire milkshake....and calculate/mix it every time.

(don't use MD5, use something deliberately slow, like bcrypt)

Horrible – storing the password

Better but still bad – storing the password hashed

Decent – storing “userid+password” hashed

Best – storing “userid+password+salt” hashed

Login: jmarks

password: g00dpassword

(salt): b00gab00ga

jmarks+g00dpassword+b00gab00ga

==HASHED==>

02f39aae85ad73e162b446e918597e89

Hey, so these hashes

They look like--

02f39aae85ad73e162b446e918597e89

What are the odds that it would look like, say..

02f39aae85ad73e162b446e000000000

It would take work to find these, right?

A bit on banks and money

What is most money "made of?" How is it stored? Coins and little green pieces of paper?

A bit on banks and money

What is most money "made of?" How is it stored? Coins and little green pieces of paper?

NOPE.

Just (trusted) lists. Ledgers in banks and such.

Usually "digital"

In fact, lists are older than “money” itself.

Not dollars, but a list somewhere that says

“Ug owes Oof two cows”

“Oof owes Grok a stick”

or more accurately

“everybody owes the king taxes”

theorem: any system that involves writing down “ownership” and “what you've paid” for is (possibly) a bank = (gamestop, even)

But, you also might want "pieces/tokens"

GOLD AND DIAMONDS HA HA HA

(wait, seriously. Why are diamonds more expensive than water?)

Bitcoin

A huge encoded/distributed online ledger/list.

Powered/driven by "mining" (which is more like a slot machine, pull the lever, power the thing, and see if you "win")

Mining Bitcoin?

- Randomly trying to find “nice looking” hashes.

  ……4E9BB99 nope.

  …….000000 yep! $$$$

Mining

When you download a bitcoin wallet program, you literally have to get a copy of every single transaction ever.

Transactions are computationally expensive.

The “Mining” also powers the “hashed transactions..eg.”

02b23 gave bf239 .005 bitcoins. I can prove it because the hash of this transaction is =>

081ee23

Add this to the chain and spread it around.

Bitcoin transaction.

- You “add your new or old hash movement” to the ledger. By making another special hash. Which is expensive.