KEEPING SECRETS IN THE DIGITAL AGE

Vaishnavi Sundararajan
Chennai Mathematical Institute

October 2018, SRM Chennai
“Three may keep a secret, if two of them are dead.”

— Benjamin Franklin
Not really possible to live without secrets.

Financial details, at the very least!

Also, state secrets, private communication etc.

A time-honoured solution: encryption.
ENCRYPTION

❖ Like putting a document in a safe
❖ Have no access to document once locked
❖ Can access only by unlocking using a key
A BRIEF HISTORY OF ENCRYPTION

- Substitution ciphers: Atbash, Mlechhita vikalpa
- Shift/transposition: Caesar cipher, Scytale
- Chinese Nüshu script: secret code for women
- Renaissance: Frequency-based substitution, Vigenère
- World Wars: One time pad, Enigma
EXAMPLE: SUBSTITUTION CIPHER

<table>
<thead>
<tr>
<th>A = B</th>
<th>H = A</th>
<th>O = O</th>
<th>V = L</th>
</tr>
</thead>
<tbody>
<tr>
<td>B = V</td>
<td>I = D</td>
<td>P = Y</td>
<td>W = P</td>
</tr>
<tr>
<td>C = G</td>
<td>J = Z</td>
<td>Q = F</td>
<td>X = U</td>
</tr>
<tr>
<td>D = Q</td>
<td>K = C</td>
<td>R = J</td>
<td>Y = I</td>
</tr>
<tr>
<td>E = K</td>
<td>L = W</td>
<td>S = X</td>
<td>Z = R</td>
</tr>
<tr>
<td>F = M</td>
<td>M = S</td>
<td>T = H</td>
<td></td>
</tr>
<tr>
<td>G = N</td>
<td>N = E</td>
<td>U = T</td>
<td></td>
</tr>
</tbody>
</table>

THIS IS A MESSAGE

HADX DX B SKXXBNK
EXAMPLE: SHIFT CIPHER

THIS IS A MESSAGE

WKLV LV D PHVVDJH
MODERN CRYPTOGRAPHY

- Schemes proposed so far are all “symmetric key”
- Both parties must use the same key to encrypt and decrypt
- Needs the key to be securely exchanged before communication! Non-trivial.

Public-key cryptography!
PUBLIC KEY CRYPTOGRAPHY

- Encryption and decryption keys different
- Easier distribution of keys: phone book style
- No need to transport “secret” keys!
- Easy encryption, harder decryption
RSA ALGORITHM

- Find three very large positive integers $e$, $d$ and $n = p^*q$ for large prime $p$, $q$ such that for all integers $m$ (with $0 \leq m < n$):
  $$((m)^e)^d \equiv m \pmod{n}$$

- Even knowing $e$ and $n$ (or even $m$), very difficult to find $d$.

- Need to factorize $n$ into $p$ and $q$! Considered a hard problem.

- Encrypting key is $e$, $n$ (public), plaintext message is $m$, decrypting key is $d$ (private).
OTHER PUBLIC KEY SCHEMES

❖ Others:

❖ Diffie-Hellman-Merkle key exchange

❖ ElGamal encryption

❖ Often slower than symmetric key cryptography.

❖ Used to exchange symmetric key securely, then symmetric encryption used for communicating!
1970s TO NOW: WHAT’S DIFFERENT?

- Internet is way more ubiquitous
- Text, email, videos — all encrypted! (Hopefully)
- Encrypted data stored on the cloud also
IMPLICATIONS

- Encrypted data is the norm, not the exception
- Schemes so far were “all or nothing”
- Now need a more granular notion of ‘secrecy’!
COMPUTING ON ENCRYPTED DATA

- School: Records are encrypted, principal has key
- Students should know their percentile
- Should not know anyone else’s marks!
FUNCTIONAL ENCRYPTION

❖ Want to be able to compute on encrypted data

❖ Functional encryption

❖ Can obtain a function of values under encryption

\[ n = f_{\text{FunE}}(\square, \square, \ldots, \square) \]

❖ Lets student find their percentile, without knowing anyone else’s marks
Hospital: Records are (naturally) encrypted, only patient and doctor have the key

Doctor wants to know the number of her patients who are above 60 years of age

Like to do the computation on the cloud itself, but records are encrypted

Nobody else should get access to this info!
HOMOMORPHIC ENCRYPTION

- Functional encryption: anyone can find out result!

- Fully homomorphic encryption

- Function on encrypted inputs; result encrypted!

\[ \square = f_{FHE}(\square, \square, \ldots, \square) \]

- Only the doctor can decrypt answer to find the actual number of her senior citizen patients.
SO FAR...

- Symmetric-key encryption
- Public-key encryption
- Functional/homomorphic encryption
Information theory (Claude Shannon, 1948)

Looking at ☐, should not get any information about underlying plaintext

Given ☐ and ☐, should not be able to tell whether same or different plaintext!
INFORMATION FLOW CONT'D.

- Symmetric/public-key encryption: Might work
- Functional/homomorphic encryption: Cannot!
  - At least the function computed is known
  - Might use that to distinguish encrypted values
  - Reveals more than plain encryption!
INFORMATION FLOW

- Want to keep “high priority” data from being visible to “low priority” users

- Think of a university database, with many people authorised to do different tasks

- Students might be authorised to see timetable, but not course grades, for example.
Boss fixes a list of salary bonuses for employees

Wants admin to add them to employee salaries

Admin who does this should remain anonymous

Must show proof of being allowed to add bonuses, without revealing identity!

Zero-knowledge proofs
ZERO-KNOWLEDGE PROOFS

✧ Want to prove a statement to someone without giving them any further knowledge about it!

✧ Want to prove “x = 1 or x = 2”

✧ Easy way: Send x, recipient figures out which it is

✧ NOT a zero-knowledge proof; recipient finds out more than x = 1 or x = 2!
WHERE'S HOMER?
WHERE’S HOMER?
WHERE'S HOMER?
WHERE'S HOMER?
ZERO-KNOWLEDGE PROOFS

- Different ways of implementation
- Depends on the underlying data
- Extend for more granular notions of info-leak

Partial-knowledge proofs: quantify how much information allowed to leak
RECAP

- Symmetric, public-key encryption
- Computing on encrypted data: functional, homomorphic encryption
- Zero/partial knowledge proofs
- Information flow analysis