Quantum Cryptography

Prof. Zeph Grunschlag
Quantum Taketh Away...

All well studied computationally-secure crypto-systems cracked with hypothetical quantum computers ($\mathbb{Q} = \text{QUANTUM}$ below)

- FACTORING $\in \mathbb{QPT} \Rightarrow$
  - Rabin cracked
  - RSA cracked
- DLOG $\in \mathbb{QPT} \Rightarrow$
  - Dlog hash function cracked
  - El-Gamal cracked
  - Diffie-Helman key exchange cracked
- Elliptic curve cryptography cracked
Hidden Kernel Problem

Amazing Fact: All of the cryptographically relevant quantum computer algorithms are specializations of the following general problem.

**Hidden Kernel Problem:** Given

- homomorpsism $\psi : G \rightarrow H$ such that
- $G$ - finitely generated commutative group
- $H$ - finite commutative group
- a quantum black-box for computing the function $U : G \times H \rightarrow G \times H$ defined by
  \[ U(g, h) = (g, \psi(g) \cdot h) \]

find a set of generators for $K = \ker(\psi) = \{ g \in G \mid \psi(g) = 1 \}$
Solubility of Hidden Kernel Problem

THM: If a QPT algorithm exists for carrying out the transformation $U$ for a given $\Psi$, then there is a QPT algorithm for solving the associated hidden kernel problem for $\Psi$. For a proof see [Nielsen & Chuang §5.4.3]

Necessary condition: For this to make sense, $U$ needs to be carried out by a quantum algorithm, so must be a unitary transformation.

LEMMA: $U$ is a unitary transformation.
DLOG $\leq$ Hidden Kernel

INPUT: Prime $p$, primitive $\alpha \in \mathbb{Z}_p^*$, any $\beta \in \mathbb{Z}_p^*$

OUTPUT: $\psi$ for which solving Hidden-Kernel gives $d = \text{dlog}_\alpha(\beta) \mod p$

Use index-calculus. Let $I = \{\text{indices mod } p-1\} = \mathbb{Z}_{p-1}^+$

- $G = I \times I$
- $H = \mathbb{Z}_p^*$
- $\psi(x, y) = \alpha^x \beta^y$
- $K = \{(x, y) \mid \alpha^x \beta^y = 1\} = \{(x, y) \mid \alpha^x \alpha^{dy} = 1\} = \{(x, y) \mid \alpha^{x+dy} = 1\} = \text{subgroup generated by } (-d, 1)$
FACTOR ≤
Hidden Kernel

Two stage proof:

1. FACTOR ≤ FIND-ORDER

2. FIND-ORDER ≤ Hidden Kernel

STAGE 1) Previously, saw that if we know of a valid RSA decryption exponent can factor. Similar proof shows that if we can find \( a, r \) such that \( a > 1, r > 0 \) and \( a^r \mod n = 1 \)
then can factor \( n \) with high probability
FIND-ORDER $\leq$ Hidden Kernel

STAGE 2) Order of $a$ is the generator of following kernel $K$:

- $G = \mathbb{Z}$
- $H = \text{image of } \psi \text{ in } \mathbb{Z}_n^*$
- $\psi(x) = a^x \mod n$
- $K = \ker(\psi) = \text{subgroup generated by } \text{ord}(a)$
...Quantum Giveth

Using Heisenberg’s uncertainty principle can design a key exchange protocol provably secure against eavesdropping.

Basic set-up:

1. Alice sends Bob photons across an insecure quantum channel eavesdropped by Eve.
2. Bob replies with measurement type list.
3. Alice returns list of valid measurement types, and tamper-check list
4. If no tampering, now have common key

Phase 1 across a “quantum channel”; Phases 2-4 classical broadcasts
AXIOMS

I. A photon **phase** may be oriented relative
   • Rectilinear Coordinates (notation: +)
   • or Diagonal Coordinates (notation: ×)

II. A photon’s **spin** within a coordinate system
    is set to 0 or 1 (the latter is really 90°)

III. Heisenberg’s uncertainty principle:
    A. can’t measure both spin and phase
       accurately simultaneously
    B. when trying to measure spin relative
       wrong phase, get {0, 1} with equal prob.
    C. photon collapses to observed result
       regardless of original state
Alice Part 1
Quantum Channel
Suppose require $k$ expected rand. secret bits.

- Alice prepares $8k$ random secret bits.

\[
\begin{array}{cccccccccccc}
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0
\end{array}
\]

- First $4k$ rand. bits represent random phases

\[
\begin{array}{cccccccccccc}
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
x & + & x & x & + & x & + & x & + & x & x
\end{array}
\]

- Second $4k$ bits represent spins
- Alice prepares and transmits $4k$ photons:

\[
\begin{array}{cccccccccccc}
x & + & x & x & + & x & + & x & + & x & x \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0
\end{array}
\]
Bob Part 2
Classical Channel

- Prepares $4k$ random phases:
  \[+ + \times + \times \times + \times \times \times +\]

- Reads Alice’s photons with respect to phase guesses:
  \[
  \begin{array}{cccccccccc}
    \times & + & \times & \times & + & \times & + & + & \times & + \\
    0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
    + & + & \times & + & \times & \times & + & \times & \times & + \\
  \end{array}
  \]

- Sends his phase-guess information to Alice:
  \[+ + \times + \times \times + \times \times \times +\]
Alice Part 3
Classical Channel

- Alice compares her phases with Bob’s keeping only agreed phases so no errors:

- Alice picks at random \( \frac{1}{2} \) of the bits for use in key and \( \frac{1}{2} \) of the bits for eavesdropping detection:

- Sends phases and actual detection bits:
Bob Part 4
Classical Channel

- Bob checks the detection bits against his corresponding measurements:

\[
\begin{array}{cccc}
+ & 1 & 1 & + \\
? & 1 & 1 & ?
\end{array}
\begin{array}{cccc}
\times & 0 \\
? & 1 & 1 & ?
\end{array}
\begin{array}{cccc}
? & 1 & 1 & ? \\
0 & 0 & ? & 0
\end{array}
\]

- If all bits agree, sends “ACCEPT” signal and uses remaining error-free bits for shared key:  

\[
\begin{array}{cccc}
? & 1 & 1 & ?
\end{array}
\begin{array}{cccc}
? & 1 & 1 & ? \\
0 & 0 & ? & 0
\end{array}
\]

\[K = \text{“100”}\]
Bob Part 4
If Eavesdropped

• If Eve observed quantum channel, when she guesses the wrong phase has 50% probability of re-transmitting the wrong bit (e.g. guesses alternating phases “+×+×...”)

<table>
<thead>
<tr>
<th>+ + + + + + + + +</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 1 0 1 0 0 0</td>
</tr>
<tr>
<td>+ × + × + × + × +</td>
</tr>
<tr>
<td>1 ? 1 0 ? ? ? ?</td>
</tr>
</tbody>
</table>

• Eve has 50% prob. of guessing wrong phase for each detection bit.
• Bob has 25% prob. of detecting wrong bit, per eavesdropped bit. If so sends “FAIL”