Hiding Signals in Quantum Random Noise

Michael Stephen Fiske
HICSS-58

Aemea Institute
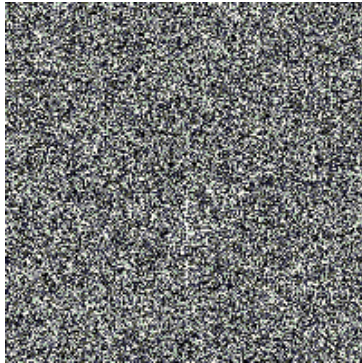
January 9, 2025

Introduction
○●○○○○○

Hiding Procedures
○○○○○○○○

Application & Testing
○○○○○

Quantum Computing
○

Summary
○○○

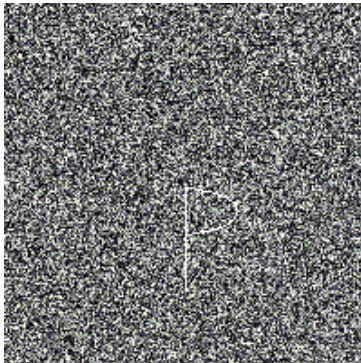## A Signal Hidden in Quantum Random Noise

The signal and noise probability distributions are identical.

A Partially Hidden Signal

The signal and noise probability distributions are slightly different.

A Detectable Signal

The signal and noise probability distributions are quite different.

Primary Contributions

Quantum random bits x_i. Heisenberg uncertainty principle.

Axiom 1: *No bias.* $P(x_i = 0) = P(x_i = 1) = \frac{1}{2}$.

Axiom 2: *Independence.* Event $H_i = \{x_1 = b_1, \ldots, x_i = b_i\}$.
Every b_j in $\{0, 1\}$. $P(x_{i+1} = 0 \mid H_i) = P(x_{i+1} = 1 \mid H_i) = \frac{1}{2}$.

- Hiding procedure: $O(n)$ fast, inexpensive, post-quantum.

- If m signal and ρ noise bits satisfy axioms 1 & 2, the signal
  can be hidden arbitrarily close to perfect secrecy ($\rho \to \infty$).

- A post-quantum key exchange with much smaller key sizes.

- Easy for signal to satisfy axioms 1 & 2. Random keys satisfy
  axioms 1 & 2. Plaintext: encrypt before hiding or embed
  signal in higher dimensional Hamming space.

Favorable Properties

- Hiding public keys hinders Mallory-in-the-middle (MITM) attacks that can attack a Diffie-Hellman exchange.

- Search complexity for hidden, public keys substantially exceeds the conjectured complexity of a public key.

- Quantum complexity is comparable to Grover's algorithm. Post-quantum Internet of Things! Less than $1.00 per device.

- Implementable with TCP/IP infrastructure & an off-the-shelf quantum random number generator (QRNG flip-flop).

- QRNG flip-flops can generate 3.3 Gigabits per second.

- Decentralization. Alice and Bob have their own QRNGs.

Related Work

- In 1550, Cardano proposed a rectangular grid for writing hidden messages. Protection was not adequate.

- Quantum cryptography (Weisner, BB84) relies on the uncertainty principle. When Eve measures a photon's polarization, it destroys the other orthogonal component. Requires polarized photons and special infrastructure to transmit polarized photons. Alice and Bob require a shared authentication secret to stop Mallory interfering with the public channel.

- Quantum secure direct communication (QSDC). QSDC claims advantages over BB84: QSDC is deterministic; every photon contributes a key bit so QSDC is more efficient; QSDC requires expensive quantum hardware and a new physical infrastructure when feasible.

Introduction
oooooo

Hiding Procedures
●oooooooo

Application & Testing
ooooo

Quantum Computing
o

Summary
ooo

A Simple Hiding Example

Signal $k_1 k_2 k_3 = 001$. $m = 3$.

Noise $r_1 r_2 r_3 r_4 r_5 r_6 r_7 = 10\ 01\ 010$. $\rho = 7$.

Map $(l_1\ l_2\ l_3) = (8\ 3\ 6)$. $n = 10$. $n = m + \rho$ always holds.

Bit $k_1 = 0$ is hidden at location 8.

Bit $k_2 = 0$ is hidden at location 3.

Bit $k_3 = 1$ is hidden at location 6.

Hidden signal $\mathcal{S}(k_1 k_2 k_3,\ r_1 r_2 r_3 r_4 r_5 r_6 r_7) = 10\ 0\ 01\ 1\ 0\ 0\ 10$.

Introduction
oooooo

Hiding Procedures
o●oooooo

Application & Testing
ooooo

Quantum Computing
o

Summary
ooo

Creating a Quantum Random Scatter Map

```
Input: n

Variables: n, j, r, t, l₁, l₂, ... lₙ.
```

$l_1 := 1 \quad l_2 := 2 \quad \ldots \quad l_n := n \qquad j := n$

while $j \geq 2$ {

    A QRNG randomly chooses r in $\{1, 2, \ldots, j\}$.

    $t := l_r$

    $l_r := l_j$

    $l_j := t$

    $j := j - 1$

}

Output: $\pi = \begin{pmatrix} l_1 & l_2 & \ldots & l_n \end{pmatrix}$

Introduction
oooooo

Hiding Procedures
oooooooo

Application & Testing
ooooo

Quantum Computing
o

Summary
ooo

Scatter Map Definitions

Map $\pi = (l_1 \ l_2 \ldots l_n)$. Signal $k_1 \ldots k_m$. Noise $r_1, r_2, \ldots r_\rho$.

Signal Locations $\{l_1 \ l_2 \ldots \ l_m\}$.

Noise Locations $\mathcal{N}(l_1 \ l_2 \ldots \ l_m) = \{1, \ldots, n\} - \{l_1, l_2, \ldots, l_m\}$.

Define scatter function $\mathcal{S} : \{0,1\}^m \times \{0,1\}^\rho \rightarrow \{0,1\}^n$.

$\mathcal{S}(k_1, \ldots, k_m, \ r_1, r_2 \ldots r_\rho) = (s_1, \ldots s_n)$.

Signal bits $s_{l_1} := k_1; \qquad s_{l_2} := k_2; \qquad \ldots \qquad s_{l_m} := k_m$.

Noise bits $s_{i_k} := r_k$. $\quad i_k$ is kth smallest number in $\mathcal{N}(l_1 \ldots l_m)$.

Introduction
oooooo

Hiding Procedures
oooo●oooo

Application & Testing
ooooo

Quantum Computing
o

Summary
ooo

Hide a Signal with Scatter Map π

Input: Signal $k_1\ k_2\ \ldots\ k_m$. Map $\pi = (l_1\ l_2 \ldots l_n)$.

Alice's QRNG creates noise $r_1\ r_2 \ldots r_\rho$. $\rho = n - m$.

Alice's map π sets $s_{l_1} = k_1\ \ldots\ s_{l_m} = k_m$.

Per $\mathcal{S}(k_1, \ldots, k_m, r_1, r_2 \ldots r_\rho)$, Alice fills in $\mathcal{S} = (s_1 \ldots s_n)$.

Alice sends $\mathcal{S}$ to Bob.

Output: Bob's π extracts $k_1 \ldots k_m$ from $\mathcal{S}$.

Introduction
oooooo

Hiding Procedures
ooooo●ooo

Application & Testing
ooooo

Quantum Computing
o

Summary
ooo

A Random Hidden Nonce Makes π Reusable

- Alice and Bob share π.

- Each transmission uses a distinct hiding map σ.

- Each time Alice's QRNG generates a new random nonce $\mathcal{N}$.

- Alice executes procedure 3 to derive σ from $\mathcal{N}$ & π.

- Alice hides her signal with map σ.

- Alice hides nonce $\mathcal{N}$, using part of π.

- Bob uses part of π to extract nonce $\mathcal{N}$ from the noise.

- Bob executes procedure 3 to derive σ from $\mathcal{N}$ & π.

- Bob uses σ to extracts Alice's signal from the noise.

Introduction
000000

Hiding Procedures
00000●00

Application & Testing
00000

Quantum Computing
0

Summary
000

Procedure 3: Randomly Generating σ

Inputs: m, n. $\pi = (l_1 l_2 \ldots l_n)$. κ, $\mathcal{N}$, j_0. Ψ is SHA-512.

$q_1 := l_1 \quad q_2 := l_2 \quad \ldots \quad q_n := l_n \quad j := j_0$.

while $j \geq 2$ {

$\quad \kappa := \Psi(\kappa) \oplus \mathcal{R}(\kappa, 8)$

$\quad \mathcal{N} := \Psi(\kappa \ \mathcal{N}) \oplus \mathcal{R}(\mathcal{N}, 8)$

$\quad r := (\mathcal{N} \mod j) + 1$

$\quad t := q_r$

$\quad q_r := q_j$

$\quad q_j := t$

$\quad j := j - 1$

}

Output: $\sigma = (q_1 \ q_2 \ \ldots \ q_m)$.

Introduction
oooooo

Hiding Procedures
ooooooo●o

Application & Testing
ooooo

Quantum Computing
o

Summary
ooo

Procedure 3 Explained at HICSS-58

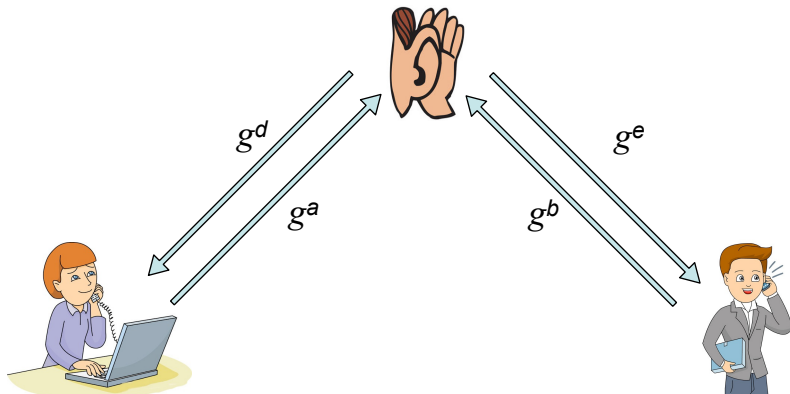## Mathematical Analysis of a Single Transmission

- If an m-bit signal & ρ bits of noise satisfy axiom 1 (unbiased) & axiom 2 (independence), our math proofs show that a one-time transmission $\mathcal{S}$ from Alice to Bob approaches perfect secrecy as ρ increases.

- Perfect secrecy: the probability that a signal $= k_1, k_2, \ldots k_m$ before Eve sees $\mathcal{S}$ remains unchanged after Eve sees $\mathcal{S}$.

- If necessary, transform the signal so it satisfies axioms 1 & 2. Good keys automatically satisfy axioms 1 & 2.

- Our proofs rely on the standard normal curve's geometry. A binomial distribution approaches the standard normal curve as $n = m + \rho$ increases. (Central Limit Theorem.)

Hiding Public Keys in Noise

- A new key exchange can hide public keys in noise.

- Hinders MITM attack on a Public Key Exchange.
  Complexity is too high for Eve.

- Implemented with the 25519 elliptic curve.[1]

- Mallory's complexity is 10^{37} for a naked 25519 public key P.
  If no auxiliary information, Mallory has no halting criteria.

- Post-quantum. Reduces key sizes. A quantum computer can
  break naked 25519 public keys in $O(n^2)$ or $O(n^3)$ steps.

---

[1] D.Bernstein.(2006) "Curve25519: new Diffie-Hellman speed records."
Public Key Cryptography.LNCS 3958. Springer. 207–228.

Introduction
oooooo

Hiding Procedures
oooooooo

Application & Testing
o●oooo

Quantum Computing
o

Summary
ooo

Hiding Hinders Mallory in the Middle Attacks

Eve and Alice share secret g^{ad}. Eve and Bob share secret g^{be}.

Introduction
oooooo
Hiding Procedures
ooooooooo
Application & Testing
oo●oo
Quantum Computing
o
Summary
ooo

A Hidden 25519 Elliptic Public Key P

Alice's hidden public key $P = $ 119 179 68 170 227 9 166 162 231 42 145 129 112 181 218 237 103 207 26 200 158 198 149 143 41 87 194 114 11 1 214 24

$\sigma(0) = 1993$. $\sigma(1) = 725$. $\sigma(2) = 405$. $\sigma(3) = 138$. $\sigma(4) = 1825$. $\sigma(5) = 1553$. $\sigma(6) = 213$. $\sigma(7) = 858$.

$n = 2048$. $m = 255$. All signal bits are blue, except first 8 bits are orange. Decimal 119 = 0111 0111

00001010010001110110101010011010000111001001010010010101111101101001001011010000100000010010110001000101
00111111110010000001010000101101110010000001011110010010110110110001110010101010101101010101010101011101
11110010110001011100101101001110011001100100100001011111101100101010111101010010001111101001111111011101
00110110010000100011000111110110001011010001100110011011101110010110010011010101011110011010110001010110
01000100010010010010111110010011110010001111111001100101111111010100001100011101011010100110011001111001
00001100001001000100011011011010101011011100001011001010010001110101010100001000010100000100010001111100
10100001111110101010010010000001111011011011000110010011011111100000010010010011011000010110110101111001
01101010101010101101110001111011000001000111110001111011111101111110110011110110111100111111011111000011
11110010001111101011101111110001000010101010111100011001011001010110010011110110111100101010100010001000
01001101010011000101000010001001011110101111011010100110010010110010010110001101010101001011101111001110

10101101111011100000010010011100110110101010010010001011011011010010100001100001101101001000110000011100
10111111100100111010000010100011110101101011011110010101101000000001010010010111001110001001101010001010
01010101011000100100110100010101000001001110001011101011011010010010110000111101010101011110101111001001
00010111111110010111000000100101010110110011000011001000111101000010011001001101010101010111111111001010
11010101010010110110010100011101100011001101010010010100001111011101001010011001001110010001110000001101
11111101011010100110111011111001111000000111111110010100010111111010110110001010101011111110111101110011
01000101000110011000100010001101001111101010111010011010010101010111010010100100011001100010000100011110
11001111111110010110011101010101101001010100101110

Complexity of Finding a 25519 Elliptic Public Key P

σ determines where P is hidden.

A random nonce hidden in the noise unpredictably changes σ each time. (Entropy Invariance.)

Every possible σ is uniformly reachable from π, based on Diehard testing of Procedure 3.

Eve knowing where P was hidden in a prior hidden transmission reveals nothing about the location of the new P.

Since there are more than 255 0s and 1s of noise, every public key P in $\{0, 1\}^{255}$ is possible.

Stops MITM attack: If Eve doesn't know π, Eve must test every possible P. That won't work.

Introduction
oooooo

Hiding Procedures
oooooooo

Application & Testing
oooo●

Quantum Computing
o

Summary
ooo

Statistical Testing of 25519 Public Keys & QR Noise

Statistical testing helps verify 25519 public keys (signal) and quantum random noise satisfy axioms 1 & 2.

```
do 80 million times   {
  a QRNG creates a 25519 private key κ.
  compute public 25519 key P from κ.
  write κ to  noise_control_file.txt
  for each bit bᵢ in byte  j of P
     write bit bᵢ in byte_j_bit_i.txt    }
```

Diehard tests on byte_j_bit_i.txt look for statistical anomalies in the ith bit of the jth byte of 25519 public keys.

Every file byte_j_bit_i.txt passed all 13 Diehard tests.

Introduction
oooooo

Hiding Procedures
oooooooo

Application & Testing
ooooo

Quantum Computing
•

Summary
ooo

Relevance to Quantum Computing

- N unsorted databased items. Classical algorithm $O(\frac{N}{2})$ steps.

- Grover's quantum algorithm takes $O(\sqrt{N})$ steps.

- Grover's algorithm requires a terminating condition.

- Scatter maps in $\mathcal{L}_{(m,n)}$ correspond to N database items.

- Eve has a terminating condition for scatter maps only if Eve has auxiliary information about σ after the scatter.

- Conjectured complexity is $O\left(\sqrt{\frac{n!}{(n-m)!}}\right)$ if Eve has a terminating condition.

- $\sqrt{\frac{8192!}{(8192-255)!}} > 10^{498}$ for $m = 255$ & $n = 8192$.

Introduction
000000

Hiding Procedures
00000000

Application & Testing
00000

Quantum Computing
0

Summary
●00

Research Summary

- A procedure hides a signal in quantum random noise.

- The locations of the signal bits randomly change each time.

- Security of the hidden signal can be made arbitrarily close to perfect secrecy.

- A new key exchange hides public keys in noise.

- Diehard tests verified that the probability distribution of 25519 public keys satisfy axioms 1 & 2.

- Our hiding procedure can be implemented with TCP/IP infrastructure and an inexpensive, off-the-shelf QRNG.

Factorial Growth vs. Exponential Growth

Set $r(n) = \frac{n!}{2^n}$. $\qquad \frac{r(k)}{r(k-1)} = \frac{k!}{2^k} \frac{2^{k-1}}{(k-1)!} = \frac{k}{2}$ $\quad$ whenever $k \geq 1$.

Hence, $r(2n) = r(n) \prod\limits_{k=1}^{n} \frac{(2n-k+1)}{2} > r(n) \, n!$ $\quad$ If $n \geq 3$, $\;\; r(2n) > n!$

```
[julia> factorial(4)
 24

[julia> 2^4
 16

[julia> function r(n)
[          r = factorial(big(n) ) / 2^(big(n))
[          return r
[          end
 r (generic function with 1 method)

[julia> r(4)
 1.5

[julia> factorial(4) / 2^4
 1.5

[julia> r(100)
 7.362140279596095642145348079335098603605904786041407178165622553205507320042596e+127

[julia> r(1000)
 3.755333903791443599585571559542306426775894026657514769644025241443938219420678e+2266
```

Future Work & Research

Future work should explore an Internet of Things (IoT)
implementation due to being low cost and post-quantum.

Based on Grover's algorithm, we anticipate Eve's quantum
complexity is $O\left(\sqrt{\frac{n!}{(n-m)!}}\right)$ when m signal bits are hidden in
$n - m$ noise bits and signal and noise satisfy axioms 1 & 2.

Future research should explore variations of Grover's algorithm to
further analyze the quantum complexity of our key exchange
hidden in noise.