Quantum encryption with certified deletion

ANNE BROADBENT, RABIB ISLAM

(UNIVERSITY OF OTTAWA)

Motivation

"I deleted the ciphertext!"

"How do I know?"

Motivation

- With a classical ciphertext, Bob cannot prove deletion to Alice
  - Bob can always make a copy of the ciphertext that can be decrypted once the key is received
- Therefore, we must consider a non-classical solution

A solution

- A quantum ciphertext?
  - No-cloning theorem: there is no map that will create an identical copy of an arbitrary quantum state
- But what would a proof of deletion look like?
- Entropic uncertainty relations: measurement in one basis can cause loss of information about what the measurement outcome in another basis would have been
- Conjugate coding (Wiesner/BB84 states) and measurements will be integral to our scheme

Previous results

Context for the idea

- [Unruh 2013] “Revocable quantum timed-release encryption”
  - Showed that a quantum encoding can be used to show “revocation”
  - Differences: CSS codes and quantum random oracles

First mention of “certified deletion”

- [Fu and Miller 2018] “Local randomness: Examples and application”
  - Verification of deletion can be done with classical interaction
  - Device-independent setting

Previous results

Independently from us:

- [Coiteux-Roy and Wolf 2019] “Proving Erasure”
  - Provable deletion using quantum encodings
  - Not about encryption schemes
  - Discussed the use of conjugate coding

Previous results

It is worthwhile to compare techniques in our scheme to those of

- [Bennett and Brassard 1984] "Quantum cryptography: Public key distribution and coin-tossing"
  - Our scheme involves less interaction
  - Still uses conjugate coding
  - Privacy amplification, error correction, entropic uncertainty relations: [Tomamichel and Leverrier 2017] "A largely self-contained and complete security proof for quantum key distribution"

Scheme: parameters

- n: length of the message
- m: number of qubits used in the quantum encoding

Scheme: key generation

- $\theta \leftarrow \{\theta \in \{0,1\}^m \mid \omega(\theta) = k\}$, where k is less than m.
  - Basis for encoding qubits
  - Content of qubits: string of length m called r
- $r_{diag} \leftarrow \{0,1\}^k$
  - Also called "check bits"
- $u \leftarrow \{0,1\}^n$
- $H \leftarrow$universal$_2$ family of hash functions
  - Domain: strings of length $m - k$; codomain: strings of length n

Scheme: encryption

- $r_{comp} \leftarrow \{0,1\}^{m-k}$
- $x \leftarrow H(r_{comp})$
- Ciphertext: r encoded in basis θ, with $msg \oplus x \oplus u$.

Scheme: decryption

- Measure qubits in basis θ to yield r, and hence r_{comp}
- Compute $H(r_{comp}) = x$.
- Compute $msg \oplus x \oplus u \oplus x \oplus u = msg$.

Scheme: delete

- Measure qubits in the Hadamard basis and obtain a certificate of deletion $\mathrm{y} \leftarrow \{0,1\}^m$

Scheme: verification

- Using θ, take the substring of the received string that corresponds to the diagonal positions of the qubits (call the result y').
- Accept if $\omega\left(r_{diag} \oplus y'\right) < \delta k$.

Error tolerance

- Linear error correcting codes can generate error syndromes
  - Corrections to a message can be made when given the syndrome of the correct message (syndrome decoding)
- In key gen: Alice samples another hash function from a different universal$_2$ family, where domain is strings of length $m-k$.
- Also samples two more strings, one the length of a syndrome, another the length of the hash function output.
- She uses one of these strings to encrypt the syndrome; she uses the other to encrypt the hash of r_{comp} made by the new hash function
- These two new values become part of the ciphertext

Error tolerance

- Process ensures correctness with high probability for a certain noise threshold
- For robustness, Bob compares the hash of his version of r_{comp} to the hash he receives from Alice
- If the hashes are not equal, the decryption went wrong

Encryption security

- Perfect ciphertext indistinguishability
  - Due to long key length

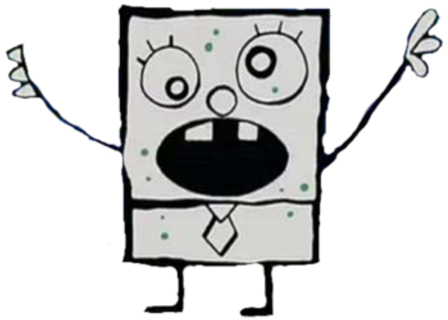
Certified deletion security: Game 1

b'

y

θ, u, H, r_{diag}

msg_0



ciphertext

b ok

Certified deletion security: Game 1

- Bob can be seen as having two goals:
  1. Determine whether his message was encrypted in the ciphertext
  2. Convince Alice that he deleted the ciphertext prior to receiving the key
- Scheme is secure if the probabilities of the following two events are negligibly close:
  1. Verification passes and Bob's guess of b is 1, in the case that Alice encrypted the string of zeros
  2. Verification passes and Bob's guess of b is 1, in the case that Alice encrypted the candidate message.

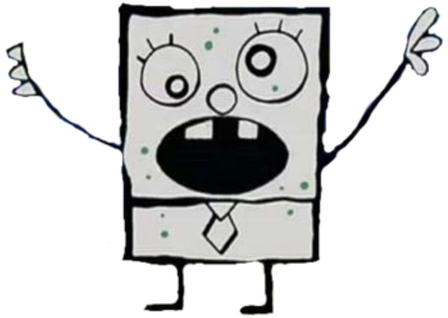
Certified deletion security: intuition

- Bob is incentivized to measure as many qubits as possible in the Hadamard basis in order to make a good proof of deletion
  - But this will lose information in the other basis, so information about r_{comp} is lost
  - A hash function has to be used in order to obtain x
- Bob also wants to measure in the computational basis to get r_{comp}
  - But check bits are encoded in Hadamard basis, and thus may be measured incorrectly
  - Incorrect measurement of checkbits will result in a proof of deletion that does not pass verification

Certified deletion security: Game 2

- Game 1 is difficult to analyze
- We developed a Game 2 which is based on an entanglement-based series of interactions
- A reduction shows that statements about Game 2 can translate into statements about Game 1
  - We thereby achieve bounds relevant to our scheme

Certified deletion security: Game 2



A

B y

B'

θ, u, H msg_0

r r_{comp} x b'

r_{diag}

b ok $msg \oplus x \oplus u \oplus x \oplus u$

Certified deletion security: similarity

- Entanglement in Game 2 corresponds to Bob's measurement in Game 1
  - Measuring everything in the Hadamard basis in Game 1 is like fully entangling A and B in Game 2 – this will give him r_{diag}
  - Measuring everything in the computational basis in Game 1 is like fully entangling A and B' in Game 2, and then measuring B' in the computational basis – this will give him r_{comp}

Entropic uncertainty relation

- Entanglement-based setting allows use of entropic uncertainty relations
- We use one from work by Tomamichel (arXiv: 1203.2142)
- Here, it can be used to describe the information trade-off that Bob is making in Game 2 using smooth min- and max-entropies.
- Takeaway: if the verification test is passed: the information that Bob has access to about r_{comp} is low with high probability

Privacy amplification

- The hash function accomplishes the task of privacy amplification
- Formalized using the Leftover Hashing Lemma from Renner
  - Lower bound on Bob's uncertainty about r_{comp} tells us how close x is to a uniformly random string from Bob's perspective
  - Bob is blocked from getting information about msg

Applications

- Protection against key leakage
- Protection against data retention
  - EU regulation 2016/679
- Everlasting security
  - Transform long-term computational assumption into a temporary one

Conclusion

- Used BB84 QKD-style logic to develop new scheme with relatively new security definition
- Potential applications
- Next steps:
  - Composability
  - Homomorphic encryption

Thank you!