Reconciliation of a Quantum-Distributed Gaussian Key

G. Van Assche, J. Cardinal and N. Cerf

arXiv e-print cs.CR/0107030

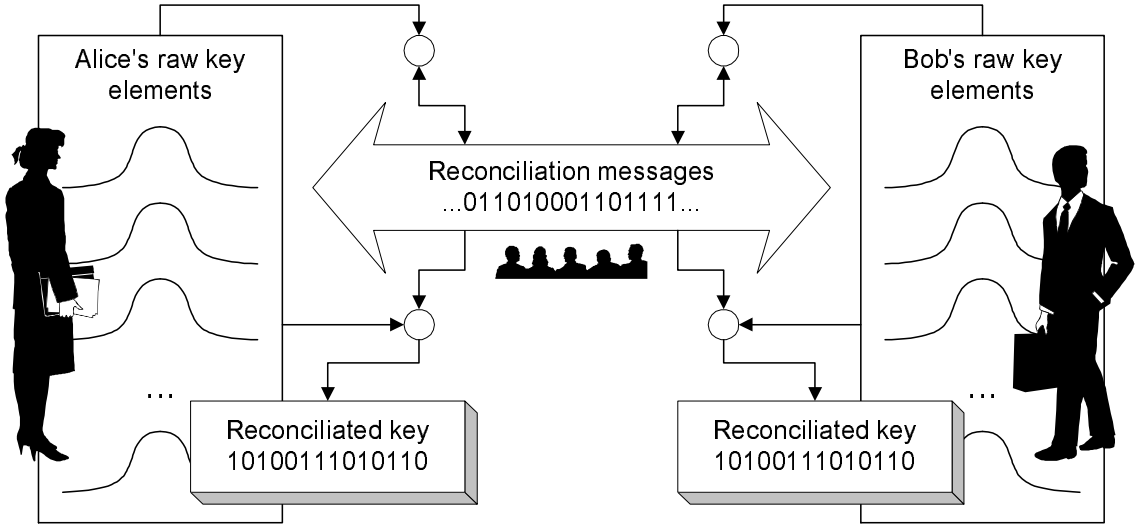
ESF workshop on
Continuous-Variable Quantum Information Processing 2002

1. Introduction

- The usual steps in quantum cryptography are:
  1. **Key distribution** over the quantum channel;
  2. **Estimation** of the channel quality, thus of Eve's information;
  3. **Reconciliation** (error correction), to get equal keys;
  4. **Privacy amplification**, to wipe out Eve's information;
  5. **Key consumption**, to make some cryptographic use of it.
- There is a need to process **continuous key elements** because some recent protocols use **continuous modulation** [1,2].
  - If Gaussian modulation is used, Alice and Bob share correlated Gaussian variables.
  - Shannon theory implies that one can transmit $I = \frac{1}{2}\log(1 + \mathrm{SNR})$ bits over a Gaussian channel. Similarly, we would like to get I private bits out of Alice's and Bob's correlated variables.

2. Philosophy

- Our reconciliation protocol produces a **discrete final key** out of **continuous key elements**.
  - This is because privacy amplification would also amplify noise in continuous key elements and because most cryptographic applications rely on a **discrete key**.
  - Note that the messages exchanged over the public authenticated channel are also discrete.



3. Sliced Error Correction

- Assume Alice has a string of Gaussian variables $X^{(1\ldots n)}$, while Bob has $X'^{(1\ldots n)}$ with each $X' = X + \epsilon$.
- Alice converts each X into m binary values $S_1(X)$, ..., $S_m(X)$, resulting in m binary strings called **slices**.
- Alice and Bob correct each slice $S_i(X)^{(1\ldots n)}$ **sequentially** for $i = 1 \ldots m$, using a binary reconciliation protocol like **Cascade** [3].
  - To estimate Alice's slice $S_i(X)^{(1\ldots n)}$, Bob uses the knowledge of both $X'^{(1\ldots n)}$ and all previously corrected slices $S_{j<i}(X)^{(1\ldots n)}$.

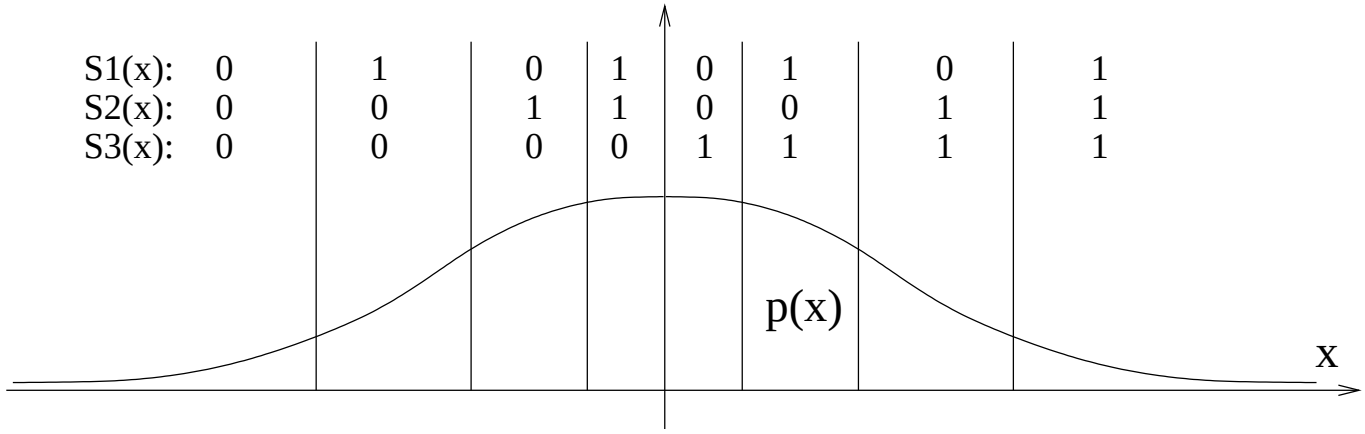
	$X^{(1)}$	$X^{(2)}$	$X^{(3)}$	...	$X^{(n-1)}$	$X^{(n)}$
S_1	0	1	1	...	0	0
S_2	1	0	0	...	0	1
...				...		
S_m	0	0	1	...	1	1

4. Properties of Sliced Error Correction

- **Asymptotic optimality:** Alice and Bob can define slices on k-dimensional key elements. When $k \to \infty$, the net amount of secret bits tends to $I(X; X')$.
- Each slice S_i has an associated **bit error rate** e_i that characterizes the correlation between Alice's and Bob's bits.
  - One can bound the amount of **disclosed information** using $I_D \leq \sum_i h(e_i)$. Of course, the disclosed information must be wiped out during the privacy amplification step.
- **Sequential correction:** The reconciliation of slice i helps to perform the reconciliation of slices $j > i$ with a lower e_j, thus with less disclosed information.
  - The first slices are almost uncorrelated between Alice and Bob (e.g., $e_1 \approx 0.5$) but are useful for the sequel.
  - Then, Alice and Bob can really extract common information from the mid-to-last slices.

5. Reconciliation of Gaussian Variables

- We cut $\mathbb{R}$ into **intervals** and we assign **bit values** to each interval for each slice. This process is optimized so as to **maximize** the net information rate, $H(S_{1\ldots m}(X)) - \sum_i h(e_i)$.
  - It is best to start reconciliating on a fine-grained scale (e.g., least significant bit of the interval number) and to end on a coarse-grained scale (e.g., most significant bit).



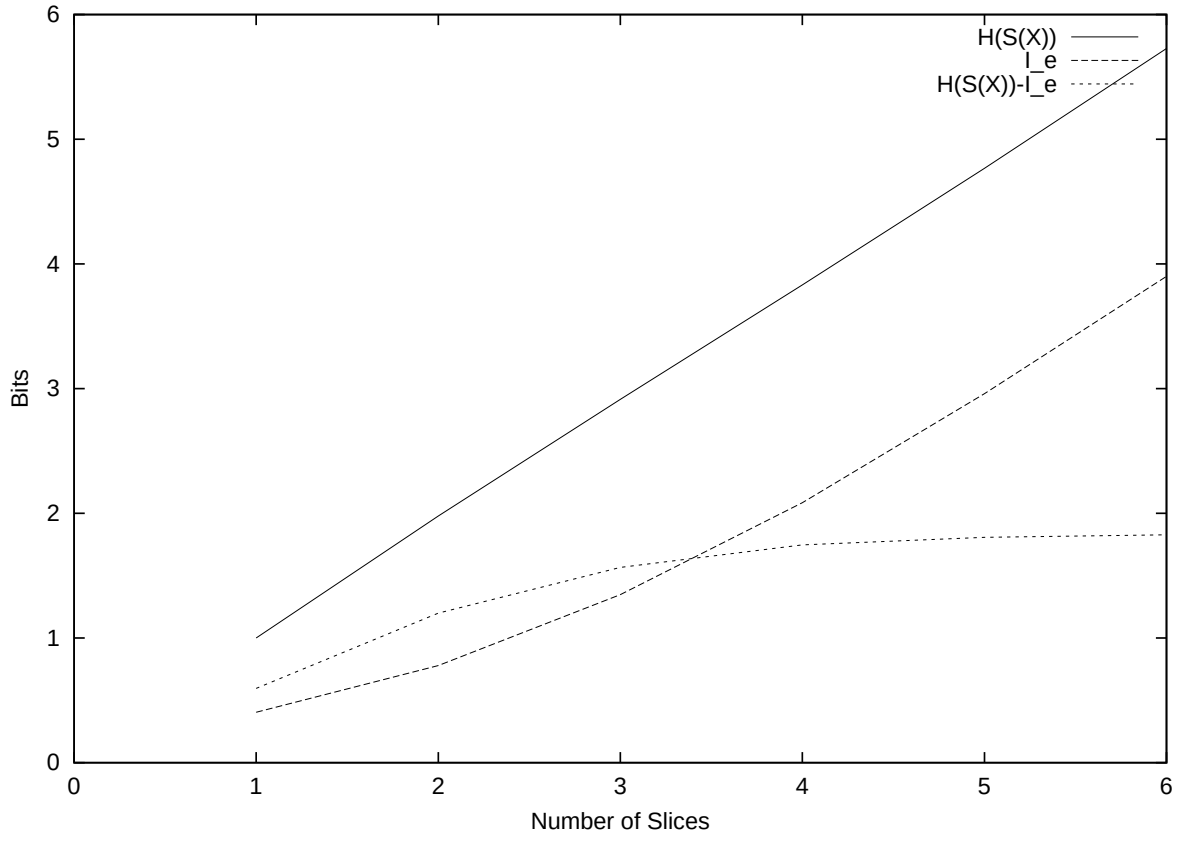
- Bob estimates Alice's bits using the **max. likelihood principle**: $\tilde{S}_i(x', S_{1\ldots i-1}(x)) = \arg\max_s P[S_i(X) = s \mid X' = x' \wedge S_{1\ldots i-1}(X) = S_{1\ldots i-1}(x)]$.

6. Numerical Experiments

- For instance, let X and X' be two Gaussian variables with SNR = 15 so that $I(X;X') = 2$ bits. Let's create $m = 5$ slices, so we divide $\mathbb{R}$ into $2^5 = 32$ intervals. After optimization, we get:
  - $e_1 = 0.482$ (least significant bit of interval number)
  - $e_2 = 0.459$
  - $e_3 = 0.184$
  - $e_4 = 8.11 \cdot 10^{-3}$
  - $e_5 = 1.06 \cdot 10^{-7}$ (most significant bit)
  - Result: $H(S_{1\ldots5}(X)) \approx 4.60$ bits produced - $\sum_i h(e_i) \approx 2.75$ bits disclosed $=$ 1.85 bits net.

7. Numerical Experiments (cont'd)

- The efficiency of the reconciliation increases with the number of slices. However, in this example, 5 slices seems a good trade-off between efficiency and computing resources.



8. Conclusions

- Our reconciliation protocol allows Alice and Bob to share a **common string of bits**, out of **correlated Gaussian variables**.
  - It is needed for quantum key distribution protocols with continuous modulation.
  - The principle is that Alice converts her Gaussian variables into bits, while Bob uses all the information available to him to disclose as few as possible information on the public channel.
  - It is **simple** and **efficient**, both in theory and in practice.

9. References

[1] N. J. Cerf, M. Lévy and G. Van Assche, Phys. Rev. A **63** 052311 (2001).

[2] F. Grosshans and P. Grangier, Phys. Rev. Let. **88** 057902 (2002).

[3] G. Brassard and L. Salvail, in Eurocrypt'93 (1993).

More references can be found in the paper.