Quantum Information and Computation, Vol. 21, No. 11&12 (2021) 0955–0973

QUANTUM DIGITAL SIGNATURES WITH SMALLER PUBLIC KEYS

BORIS ŠKORIĆ
Department of Mathematics and Computer Science, Eindhoven University of Technology
5600MB Eindhoven, The Netherlands

Received January 5, 2021
Revised May 23, 2021

We introduce a variant of quantum signatures in which nonbinary symbols are signed instead of bits. The public keys are fingerprinting states, just as in the scheme of Gottesman and Chuang [1], but we allow for multiple ways to reveal the private key partially. The effect of this modification is a reduction of the number of qubits expended per message bit. Asymptotically the expenditure becomes as low as one qubit per message bit. We give a security proof, and we present numerical results that show how the improvement in public key size depends on the message length.

Keywords: quantum cryptography

Communicated by: S Braunstein & H Zbinden

1 Introduction

1.1 Quantum signatures; unconditional security

Digital signatures and Public Key Infrastructure (PKI) form the cornerstone of our 'open' digital world; they allow people to verify the origin and integrity of data received from new communication partners, in an almost entirely non-interactive ('offline') way and based merely on a small number of public keys stored locally.

In a typical signature scheme each user owns a private key s, which is kept secret, and the related public key p, which is published. The public key is easily computed from the private key, but the reverse computation is difficult because it involves a hard problem such as factorisation, discrete logarithms, learning with errors, or a shortest vector problem. Signing is an operation that takes as input s and a message m, and outputs a signature z. Verification has the triplet (m, p, z) as input, and produces a yes/no output, where 'yes' indicates that the signature z is consistent with m and p. A signature scheme has to satisfy three security properties: (i) Unforgeability. For someone who does not hold s it is prohibitively difficult to create such a valid triplet; (ii) Non-repudiation. If a valid triplet (m, p, z) is observed, then the party associated with p cannot deny that it has created the triplet and hence endorses the message m; (iii) Transferability. If a verifier accepts a signature, he is confident that any other verifier will also accept it.

The main weakness of digital signature schemes is their reliance on a difficult computational problem, whose hardness is impossible to prove. For this reason alternative schemes have been studied [2, 3, 4] that offer *unconditional security*. These works have a number of disadvantages

in common. They have to work with a fixed set of participants, and they involve a large amount of communication. Furthermore, they require either a trusted third party or secret channels between pairs of participants.

Gottesman and Chuang [1] introduced *quantum digital signatures*, which are unconditionally secure and alleviate some of these disadvantages. The main idea is based on the observation that state preparation can be seen as a one-way function. Consider a prover Peggy who gives a quantum state to a verifier Victor. It is easy for Peggy to put a huge amount of information into a quantum state but impossible for Victor to extract all of it. It is also straightforward for Peggy to convince Victor that she knows exactly what the state is. From this unconditionally secure one-way function one can then build a Lamport-like [5] signature scheme. In the Gottesman-Chuang scheme [1] (which we will abbreviate as 'GC01') the private key is the classical data that Peggy puts into quantum states; the thus produced states are the public key. Multiple instances of the public key are allowed to exist, and these are given to the verifiers. It does not have to be fixed beforehand who the verifiers are, and they do not have to communicate beforehand; this flexibility is the main advantage of quantum signatures over the classical unconditionally-secure schemes.

In GC01 it is implicitly assumed that there exists some mechanism by which the verifiers can trust that the quantum states they receive ultimately originate from Peggy. This mechanism must not rely on standard PKI with its computational assumptions but e.g. on trusted point-to-point contacts. The complications of such a key transport mechanism are a disadvantage compared to ordinary PKI. A further disadvantage is of course the need for quantum memory at the verifiers' side, and for quantum channels.

In 2014–2015 several versions of quantum signatures were introduced [6, 7] that do not need quantum memory. However, they have the disadvantage that all recipients of the public key[a] need to participate in the distribution stage of the protocol.

A review of quantum signatures was given in [8].

1.2 Our contribution

We introduce a new variant of Gottesman-Chuang like quantum signatures (with quantum memory) in which Peggy is able to 'open' a public key in multiple ways, thus signing a non-binary symbol instead of a bit. Our public-key qudits are fingerprinting states [9, 10]. Our digital signature reveals only *a substring of the full string embedded in the public key*; the substring can be chosen in multiple different ways. We show that this method reduces the amount of public-key material required for the signing of a message. For the sake of efficiency our scheme uses the idea suggested in [1] to work with codewords instead of repeated public keys, but it does so with *non-binary* symbols.

The price to pay for revealing only partial information is that there is now a nonzero error probability when verifying a legitimate qudit (compared to zero in [1]), and furthermore forgery becomes slightly easier. Nevertheless, the overall tradeoff between security and efficiency works in our favour: at a given level of security (expressed as the gap between Peggy's and the adversary's success probability to open a qudit) our scheme spends fewer qubits per signed message bit than [1], approximately $1 + \frac{\log(T \log T)}{\log S}$, where T is the number of verifiers and S is the size of the alphabet (see Section 4.1). Asymptotically the size of the public key

[a] Confusingly refered to as 'signature'.

approaches as little as one qubit per signed message bit. In contrast, GC01 needs at least $\approx \log(T \log T)$.

The outline of this paper is as follows. In the preliminaries (Section 2) we introduce notation and list a number of useful lemmas. We briefly recapitulate the GC01 scheme [1] and fingerprinting states [9]. In Section 3 we look at the relation between non-repudiation on the one hand and correctness and security against forgery on the other hand. We discuss the difference between the true reject and false reject probability as a performance indicator. In Section 4 we look at GC01 in more detail and derive a lower bound on the number of qubits spent per signed message bit. In Section 5 we introduce our scheme, and in Section 6 we present the analysis. We summarize in Section 7.

2 Preliminaries

2.1 Notation, attacker model, and security definitions

Notation. There are T verifiers. We write d for the dimension of the public-key Hilbert space. We use the notation $[d] = \{0, \ldots, d-1\}$. A private key is a string $k \in \{0,1\}^d$. Let $\mathcal{I} \subset [d]$ be a subset. We write $k_{\mathcal{I}}$ for the substring $(k_i)_{i\in\mathcal{I}}$ where ordering of $\mathcal{I}$ is applied. The complement of $\mathcal{I}$ is denoted as $\mathcal{I}^{\rm c} = [d] \setminus \mathcal{I}$.

Our scheme signs non-binary symbols in an alphabet $\mathcal{S}$ of size S. We write $\mathcal{S} = \{0, \ldots, S-1\}$. The Hamming weight of a binary string x is denoted as $|x|$. The bitwise XOR of binary strings x and y is written as $x \oplus y$.

The notation h stands for the binary entropy function $h(p) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}$.

Attacker model. The adversary has unlimited (quantum) computing power, as well as measurement and state preparation equipment that is entirely without noise. The adversary has no access to the labs of the other parties (e.g. through side channels).

Security definitions.

We work in the following setting. Let $\mathcal{H}$ be a Hilbert space. Peggy has a private key k which is a classical string. She uses k to create T copies of a public key $|P_k\rangle \in \mathcal{H}$.[b] A signature of a classical message $m \in \mathcal{M}$ is a classical string $r = \mathsf{Sign}(k, m)$, $r \in \mathcal{R}$ which is computed as a function of k and m. Signature verification is an algorithm Verif that acts on a state $|P\rangle \in \mathcal{H}$, a message $m \in \mathcal{M}$ and a string $r \in \mathcal{R}$, yielding outcome $v = \mathsf{Verif}(|P\rangle, m, r) \in \{\texttt{REJ}, \text{1-}\texttt{ACC}, \text{0-}\texttt{ACC}\}$. Here `REJ` stands for rejection; 1-`ACC` means that Victor considers the signature to be valid, and that he is confident that any other verifier will also consider it to be valid; 0-`ACC` means that Victor considers the signature to be valid, but is not sure about other verifiers.

Definition 1 (Correctness) *We say that the signature scheme is* correct with error ε *if*

$$\forall_{k,m} \;\; \Pr\Big[\mathsf{Verif}\Big(|P_k\rangle, m, \mathsf{Sign}(k,m)\Big) = \textit{1-}\texttt{ACC}\Big] \geq 1 - \varepsilon. \tag{1}$$

Definition 2 (Security against forgery) *Let $k \in \{0,1\}^d$ be generated randomly, and let $|P_k\rangle$ be the corresponding public key state. Consider an adversary who has access to $|P_k\rangle^{\otimes T}$, chooses one message $m \in \mathcal{M}$ and receives the signature $r = \mathsf{Sign}(k, m)$. The adversary then*

[b] We assume that there is a mechanism for distributing public keys. In this respect we do not deviate from the assumptions made in [1].

outputs a pair (m', r')*, with* $m' \in \mathcal{M}$*,* $r' \in \mathcal{R}$*. We call the signature scheme* ε*-secure against forgery if*

$$\Pr\Big[m' \neq m \ \wedge \ \mathsf{Verif}(|P_k\rangle, m', r') \neq \mathtt{REJ}\Big] \leq \varepsilon. \tag{2}$$

Here the probability is taken over the random k*, the adversary's random choices, and the nondeterministic outcome of* Verif*.*

Definition 3 (Non-repudiation heuristic) *Let malicious Peggy pick any state* $|\Psi\rangle \in \mathcal{H}$*, any message* $m \in \mathcal{M}$*, and any string* $r \in \mathcal{R}$*; these are given to the* T *verifiers. Let each verifier independently execute* $\mathsf{Verif}(|\Psi\rangle, m, r)$*. Let* $N_{\mathtt{1ACC}}, N_{\mathtt{REJ}}$ *denote the number of verifiers that get result 1-*ACC*,* REJ *respectively. We call the signature scheme* ε*-secure against repudiation if*

$$\forall_{\Psi,m,r} \ \ \Pr[N_{\mathtt{1ACC}} \geq 1 \wedge N_{\mathtt{REJ}} \geq 1] \leq \varepsilon. \tag{3}$$

Def. 3 does not allow malicious Peggy to hand out different states to different verifiers, in contrast to the repudiation attacker model in GC01 which allows more general (entangled) states that pass swap tests. Hence Def. 3 should be seen as a security heuristic and not a full security definition.

2.2 Tail bounds

Lemma 1 *Let* $r \leq \frac{n}{2}$*. The following inequalities hold,*

$$\frac{2^{nh(r/n)}}{\sqrt{8r(1-r/n)}} \leq \sum_{k=0}^{r} \binom{n}{k} \leq 2^{nh(r/n)}. \tag{4}$$

Proof: For the first inequality see e.g. p.121 of [11]. The second inequality is a special case of Chernoff-Hoeffding with probability parameter 1/2. □

Lemma 2 (Chernoff bound) *Let* $X = \sum_i X_i$ *with* $X_i \in \{0,1\}$ *independent random variables. Let* $\mu = \mathbb{E}X$*. Then for any* $\delta > 0$ *it holds that*

$$\Pr[X \geq \mu + \mu\delta] \leq e^{-\frac{1}{2+\delta}\delta^2\mu}. \tag{5}$$

$$\Pr[X \leq \mu - \mu\delta] \leq e^{-\frac{1}{2}\delta^2\mu}. \tag{6}$$

2.3 The Gottesman-Chuang scheme [1]

We briefly summarize the efficient version of GC01, using codewords, as presented in Section 8 of their paper.

The message to be signed is $x \in \{0,1\}^K$. It is encoded into a codeword $c \in \{0,1\}^N$. The distance of the code is M. The private key is $k = (k_j^0, k_j^1)_{j=1}^N$, with $k_j^{0/1} \in \{0,1\}^L$. The public key consists of $2N$ d-dimensional qudit states $(|P_j^0\rangle)_{j=1}^N$, $(|P_j^1\rangle)_{j=1}^N$, with $|P_j^b\rangle = |F(k_j^b)\rangle$, where F denotes some method of embedding the string k_j^b into the qudit. There are T copies of the public key. The parameter δ, which depends on the embedding method and the dimension of the Hilbert space, is defined as

$$\delta = \max_{k,k':k\neq k'} \Big|\langle F(k')|F(k)\rangle\Big|. \tag{7}$$

Peggy's signature of the string x consists of the private keys $(k_j^{c_j})_{j=1}^N$. Signature verification is done by projecting the state $|P_j^{c_j}\rangle$ in possession of the verifier onto the direction $|F(k_j^{c_j})\rangle$,

for each $j \in \{1, \cdots, N\}$, and counting the number of '0' results ('z'). There are two threshold parameters, $z_{\rm acc}, z_{\rm rej}$, with $z_{\rm acc} < z_{\rm rej} \leq M/2$. If $z \leq z_{\rm acc}$ then the result of the verification is `1-ACC`; if $z \geq z_{\rm rej}$ then it is `REJ`. In between, the result is `0-ACC`.
Regarding forgery the following result was shown. An adversary who holds all T copies of the public key can learn no more than $T \log d$ bits of information about the private key $k_j^{c_j} \in \{0,1\}^L$. The forgery probability for a single qudit is therefore upper bounded by the following value,

$$p_{\rm forge1}^{\rm GC} = \frac{1}{2^{L-T\log d}} + (1 - \frac{1}{2^{L-T\log d}})\delta^2. \tag{8}$$

2.4 Fingerprinting states

Quantum fingerprinting was introduced in [9] as a way to do string equality testing based on a representation that is exponentially smaller than the classical string. Let $x \in \{0,1\}^d$, $x = (x_j)_{j=0}^{d-1}$. Let $|0\rangle, \ldots, |d-1\rangle$ be an orthonormal basis of a d-dimensional Hilbert space $\mathcal{H}$. The *fingerprinting state* $|\mu(x)\rangle$ for the string x is the following state in $\mathcal{H}$,

$$|\mu(x)\rangle \stackrel{\rm def}{=} \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} (-1)^{x_j} |j\rangle. \tag{9}$$

These states have been used for various other purposes, e.g. noise-tolerant QKD [12]. It was proposed by Gottesman and Chuang to use fingerprinting states as the embedding mechanism 'F' in their quantum signature scheme.

Lemma 3 *Let $x, y \in \{0,1\}^d$. The inner product of the two fingerprint states $|\mu(x)\rangle$ and $|\mu(y)\rangle$ is given by*

$$\langle \mu(y)|\mu(x)\rangle = 1 - 2\frac{|x \oplus y|}{d}. \tag{10}$$

Proof: From the definition (9) we get $\langle \mu(y)|\mu(x)\rangle = \frac{1}{d}\sum_{j,\ell=0}^{d-1}(-1)^{y_\ell + x_j}\langle \ell | j\rangle = \frac{1}{d}\sum_{j=0}^{d-1}(-1)^{y_j+x_j}$ $= \frac{1}{d}\sum_{j=0}^{d-1}(1 - 2y_j \oplus x_j) = \frac{1}{d}(d - 2|y \oplus x|)$. □

3 Figures of merit

Given a state $|P\rangle \in \mathcal{H}$, a message $m \in \mathcal{M}$ and an alleged signature $r \in \mathcal{R}$, let $Q_{\rm R}, Q_0, Q_1$ denote the probability that $\mathsf{Verif}(|P\rangle, m, r)$ yields outcome `REJ`, `0-ACC`, `1-ACC` respectively. Let $N_{\texttt{REJ}}, N_{\texttt{0ACC}}, N_{\texttt{1ACC}}$ denote the number of verifiers who get those outcomes.

Lemma 4 *The repudiation probability (3) can be expressed as*

$$P_{\rm repud} \stackrel{\rm def}{=} \Pr[N_{\texttt{1ACC}} \geq 1 \wedge N_{\texttt{REJ}} \geq 1] = 1 - (1-Q_{\rm R})^T - (1-Q_1)^T + Q_0^T. \tag{11}$$

Proof: The outcome for each of the T verifiers is independent and follows the same distribution $(Q_{\rm R}, Q_0, Q_1)$. The left hand side of (11) can be written as a partial sum over the multinomial probability distribution, $\sum_{a=1}^{T-1}\sum_{b=1}^{T-a}\frac{T!}{a!b!(T-a-b)!}Q_{\rm R}^a Q_1^b Q_0^{T-a-b}$, which can be rewritten as $\sum_{a=1}^{T-1}\binom{T}{a}Q_{\rm R}^a \sum_{b=1}^{T-a}\binom{T-a}{b}Q_1^b Q_0^{T-a-b} = \sum_{a=1}^{T-1}\binom{T}{a}Q_{\rm R}^a[(Q_0+Q_1)^{T-a} - Q_0^{T-a}]$. Finally we use the binomial sum rule twice, subtracting the $a=0$ and $a=T$ terms. □

Corollary 1 *The following inequalities hold*

$$P_{\rm repud} \leq 1 - \Big(1 - \min(Q_{\rm R}, Q_1)\Big)^T \leq 1 - \Big(\max(Q_{\rm R}, Q_1)\Big)^T \tag{12}$$

$$P_{\rm repud} \leq T \cdot \min(Q_{\rm R}, Q_1). \tag{13}$$

Proof: The first inequality in (12) is obtained from (11) by using $Q_0^T \leq (1-Q_{\rm R})^T$ and $\overline{Q_0^T} \leq (1-Q_1)^T$. The second one follows from $Q_1 + Q_{\rm R} \leq 1$. The inequality (13) follows from the first expression in (12) by using $(1-x)^T \geq 1 - Tx$. □

Lemma 5 *Consider a prover who hands out a state in $\mathcal{H}$ which is indeed the public key $|P_k\rangle$ for some private key k. If a signature scheme is correct with ε_1-error and is ε_2-secure against forgery, then for any $m \in \mathcal{M}$, $r \in \mathcal{R}$ the repudiation probability $P_{\rm repud}$ is upper bounded by $T \cdot \max(\varepsilon_1, \varepsilon_2)$.*

Proof: We distinguish between two cases, (i) r is a correct signature, and (ii) r is not a correct signature. For the first case we use (13) to write $P_{\rm repud} \leq TQ_{\rm R} \leq T(1-Q_1)$, with $Q_1 \geq 1-\varepsilon_1$. Similarly, in the second case we use (13) to write $P_{\rm repud} \leq TQ_1 \leq T(1-Q_{\rm R})$, with $Q_{\rm R} \geq 1-\varepsilon_2$. □

Note that Lemma 5 does not imply non-repudiation as defined in Def. 3. The reason is that correctness and security against forgery are defined only for quantum states $|P\rangle$ which are a proper public key $|P_k\rangle$ for some private key k, whereas the definition of non-repudiation allows Peggy to distribute *any* state in $\mathcal{H}$.

On the other hand, Lemma 5 provides a guideline on how the correctness error and the security against forgery should be tuned if one aims at a certain level of non-repudiation.

We add a superscript 'genuine' or 'forgery' on the probabilities $Q_{\rm R}, Q_0, Q_1$ to distinguish between the two cases.

In the schemes that we focus on in this paper, we have $\mathcal{H} = \mathcal{H}_1^{\otimes N}$. The $\mathcal{H}_1$ is referred to as a qudit space. The verifier performs a binary projective measurement on each of the N individual qudits, e.g. the projection onto $|F(k_j^{c_j})\rangle$ in GC01 (Section 2.3), and gets a tally $z \in \{0, \ldots, N\}$ of how many errors occur. (By 'error' we mean that a qudit does not pass verification, i.e. the projection yields '0'.) Let G denote the per-qudit error probability in case of a *genuine* signature, and J in case of a mismatch between the signature and the quantum state in $\mathcal{H}_1$. The relevant quantity for the correctness property is the error tally in case of a genuine signature,

$$Q_1^{\rm genuine} = \Pr[Z \leq z_{\rm acc} | \text{genuine}] = \sum_{z=0}^{z_{\rm acc}} \binom{N}{z} G^z (1-G)^{N-z}. \tag{14}$$

If a lower bound $Q_1^{\rm genuine} \geq 1 - \varepsilon_c$ can be proven, then the scheme is 'correct with error ε_c' as specified in Def. 1. For the security against forgery the relevant quantity is

$$Q_{\rm R}^{\rm forgery} = \Pr[Z \geq z_{\rm rej} | \text{forgery}]. \tag{15}$$

If a lower bound $Q_{\rm R}^{\rm forgery} \geq 1 - \varepsilon_f$ can be proven, then the scheme is 'ε_f-secure against forgery' as specified in Def. 2. Unfortunately, a relation such as (14) does not necessarily exist between $Q_{\rm R}^{\rm forgery}$ and J, as we will see in Section 6, because in case of a forgery not all positions $1, \ldots, N$ have to be a mismatch. However, it is clear that $J > G$ is a necessary condition in order to have a working signature scheme. Let N' ($N' < N$) be the number of positions where a forgery causes a mismatch in the quantum state. The error tally $z^{\rm forgery}$ is peaked around $N'J + (N-N')G$, whereas $z^{\rm genuine}$ is peaked around NG. In order to distinguish between genuine signatures and forgeries, the scheme must have a significant distance between the thresholds $z_{\rm rej}$ and $z_{\rm acc}$, which implies the condition $N'J + (N-N')G > NG \Leftrightarrow N'(J-G) > 0$.

Table 1. *Notation.*

α	$= (d-\ell)/d$. Relative size of non-revealed substring.
β	In GC01: bit error rate that can be corrected by the code
c	codeword. $c \in \mathcal{S}^N$
d	Dimension of the Hilbert space. String length.
δ	In GC01: $\|\langle P(k)\|P(k')\rangle\| \leq \delta$
φ	Small parameter. $\varphi = d^{-1}T \log d$
G	single-qudit false reject probability
$\mathcal{I}$	Set containing the indices of non-revealed positions. $\|\mathcal{I}\| = d - \ell$.
J	single-qudit true reject probability
k	Private key. In our scheme $k \in \{0,1\}^d$.
κ	Substring of private key. $\kappa \in \{0,1\}^\ell$
K	message length
L	Private key size in GC01.
ℓ	number of revealed positions
$\|\mu(z)\rangle$	Fingerprinting state for string $z \in \{0,1\}^d$.
n	size of public key in qubits. $n = \log_2 d$
N	codeword length
$\|P\rangle$	Public key.
$\|\psi\rangle$	state derived from index set $\mathcal{I}$ and substring κ
Q_{R}, Q_0, Q_1	probability of Reject, ACC-0, ACC-1
s	Symbol to be signed. $s \in \mathcal{S}$.
S	Alphabet size. $S = 1/\alpha$.
$\mathcal{S}$	Alphabet. $\|\mathcal{S}\| = S$. $\mathcal{S} = \{0, \ldots, S-1\}$.
T	Number of verifiers. (Number of copies of each public key)
θ	bit error rate that can be corrected
x	Message. $x \in \mathcal{S}^K$.

The distance grows with N'. A large distance is necessary in order to reduce the overlap between the right tail of z^{genuine} and the left tail of z^{forgery}.
Because of the structure discussed above, we adopt the '**gap**' $J - G$ as one of the central figures of merit. The other figures of merit are the length of the code (N) and the total size of the public key expressed in qubits ($N \log d$).

4 Analysis of the Gottesman-Chuang scheme

4.1 Gottesman-Chuang with fingerprinting states

We present an analysis of the GC01 scheme that explicitly writes out a number of parameters that were not worked out in detail in [1]. We consider the efficient implementation with codewords and fingerprinting states. We look only at long-message asymptotics.
The quantities of interest are (i) The number of qubits spent on signing a whole message; (ii) the value of the parameter δ; (iii) the ensuing single-bit forgery success probability, and (iv) the minimum codeword length (number of public keys) required to upper-bound the overall message forgery probability to some fixed value.
In particular, the number of spent qubits and the single-bit forgery probability are important as a benchmark for evaluating the performance of our own scheme.

The number of spent qubits per message bit.
The error-correcting code is a binary code ('$\mathcal{C}_1$') with message size K, codeword size N, and distance M. Then $\beta \stackrel{\text{def}}{=} \frac{M}{2N}$ is the error rate that can be corrected. Asymptotically for large messages it holds[c] that $\frac{K}{N} \approx 1 - h(\beta)$.
The number of public keys required for signing the codeword is $2N$. Each public key state comprises $\log d$ qubits. Hence the number of qubits involved in signing a K-bit message is $2N \log d$.

$$\text{GC01 \#qubits per message bit} = \frac{2N \log d}{K} \approx \frac{2 \log d}{1 - h(\beta)}. \tag{16}$$

Note that only half of the public keys are 'opened'. The expenditure of qubits may be halved if there is a way of re-using the unused public keys.

The parameter δ.
A code $\mathcal{C}_2$ is used in the embedding, with message length L and codeword length d. The key $k \in \{0,1\}^L$ ($L < d$) is embedded in the public key as as the fingerprinting state of a d-bit codeword in $\mathcal{C}_2$. We denote the correctable error rate of this code as γ, with (again asymptotically) $\frac{L}{d} \approx 1 - h(\gamma)$. In order to compute the parameter δ as defined in (7) we consider two keys k, k' which differ only by a single bit flip. Their codewords differ in $2\gamma d$ bits. From Lemma 3 it then follows that

$$\delta = 1 - 4\gamma. \tag{17}$$

The single-bit forgery probability.
Substitution of (17) into (8) yields

$$\begin{aligned} p_{\text{forge1}}^{\text{GC}} &= (\frac{1}{2})^{d[1-h(\gamma)]-T\log d} + \Big\{1 - (\frac{1}{2})^{d[1-h(\gamma)]-T\log d}\Big\}(1-4\gamma)^2 \\ &= 1 - 8\gamma(1-2\gamma)\Big\{1 - (\frac{1}{2})^{d[1-h(\gamma)]-T\log d}\Big\} \end{aligned} \tag{18}$$

where we have expressed L in terms of d and γ.
The error probability J introduced in Section 3 equals $1 - p_{\text{forge1}}$. Furthermore, in GC01 the error probability G vanishes, $G^{\text{GC}} = 0$. Hence the 'gap' figure of merit is given by

$$J^{\text{GC}} = 8\gamma(1-2\gamma)\Big\{1 - (\frac{1}{2})^{d[1-h(\gamma)]-T\log d}\Big\}. \tag{19}$$

Note that the condition $L > T \log d$ has to hold, which translates to $\frac{d}{\log d} > \frac{T}{1-h(\gamma)}$. Hence there is a lower bound $d_{\min}^{\text{GC}}$ on the dimension of the Hilbert space, dictated mostly by the parameter T. For small γ this bound is

$$d_{\min}^{\text{GC}} \approx T \log T. \tag{20}$$

Substitution of (20) into (16) yields the following approximation for the minimum size of the public key,

$$\text{Small } \gamma: \quad \text{GC01 \#qubits per message bit} \gtrapprox \frac{2\log(T\log T)}{1 - h(\beta)}. \tag{21}$$

[c] Ref.[13] gives the following result for the length of the syndrome, $nh(\beta) + \sqrt{n}\Phi^{\text{inv}}(10^{-6})\sqrt{\beta(1-\beta)}\log\frac{1-\beta}{\beta}$, where Φ is defined as $\Phi(z) \stackrel{\text{def}}{=} \int_z^\infty (2\pi)^{-1/2}\exp[-x^2/2]\mathrm{d}x$.

Minimum codeword length.

Let $Q_{\mathrm{R}}^{\mathrm{forgery}}$ be a target value that we want to achieve regarding the forgery detection probability for a whole message. The easiest forgery is to flip a single bit in the message x. This causes $2\beta N$ flips in the codeword c. The forger has probability of at most $p_{\mathrm{forge1}}^{\mathrm{GC}}$ to repair a flip. Hence the expected number of leftover 'wrong' bits counted by the tally z is $\mathbb{E}z = 2\beta N(1 - p_{\mathrm{forge1}}^{\mathrm{GC}}) = 2\beta N J^{\mathrm{GC}}$. The threshold z_{rej} has to be set as

$$z_{\mathrm{rej}} = 2\beta N J^{\mathrm{GC}} - 2\sqrt{\beta N J^{\mathrm{GC}}}\sqrt{\ln(1 - Q_{\mathrm{R}}^{\mathrm{forgery}})^{-1}} \tag{22}$$

(or smaller). From Lemma 2 it follows that this setting indeed yields the correct bound on the forgery probability. The requirement $z_{\mathrm{rej}} > 0$ leads to the condition $N \geq N_{\mathrm{min}}^{\mathrm{GC}}$, with

$$N_{\mathrm{min}}^{\mathrm{GC}} = \frac{\ln(1 - Q_{\mathrm{R}}^{\mathrm{forgery}})^{-1}}{\beta J^{\mathrm{GC}}}. \tag{23}$$

Choosing a small γ on the one hand reduces the dimension d, but on the other hand increases $N_{\mathrm{min}}^{\mathrm{GC}}$. Similarly, setting β small reduces the number of qubits spent per message bit (16) but increases $N_{\mathrm{min}}^{\mathrm{GC}}$. For $\gamma \ll 1$ we have $N_{\mathrm{min}}^{\mathrm{GC}} \propto \frac{1}{\beta\gamma}$.

4.2 Gottesman-Chuang with low-dimensional embedding

In [1] the possibility was mentioned of embedding $k \in \{0,1\}^L$ into a single qubit ($d = 2$). Though possible, it has the drawback that the forgery error probability J gets exponentially close to 1, namely $J = \mathcal{O}(2^{-L})$. The L is lower bounded as $L > T \log d = T$, which yields $J = \mathcal{O}(2^{-T})$.

We briefly comment on the possibility of embedding k into a Hilbert space of dimension d larger than 2 but much smaller than (20). Optimal spreading of states is equivalent to distributing 2^L points equally over a hypersphere of dimension $\sigma = 2d - 2$. Each point dominates a σ-dimensional solid angle of order 2^{-L}, and hence the angle between neighbouring points is $\mathcal{O}(2^{-L/\sigma})$. The parameter δ is the cosine of this angle. Substitution into (8) gives $1 - p_{\mathrm{forge1}} = \mathcal{O}(2^{-2L/\sigma}) = \mathcal{O}(2^{-L/[d-1]}) < \mathcal{O}(2^{-\frac{T \log d}{d-1}})$. At fixed small d the distinction between Peggy and the attacker is exponentially small in T.

Because of the exponentially small J value in low-dimensional embedding, we will use the fingerprinting-based version of GC01 as our benchmark.

5 Our protocol for signing a nonbinary string

5.1 Intuition

We propose a scheme that is similar to the Gottesman-Chuang scheme with codewords and fingerprinting states, but which allows Peggy to 'open' a public key in S different ways. The choice how to open corresponds to signing a symbol $s \in \mathcal{S} = \{0, \ldots, S-1\}$. Peggy's private keys are $(k^i)_{i=1}^N$ with $k^i \in \{0,1\}^d$. A public key $|P_i\rangle$ consists of the fingerprinting state $|\mu(k^i)\rangle$, i.e. without the use of an error-correcting code '$\mathcal{C}_2$' in the embedding.

Peggy 'opens' the public key $|P_i\rangle$ by revealing a length-ℓ substring of k^i. The choice of non-revealed positions encodes the symbol that is to be signed. The verification step is to project $|P_i\rangle$ onto the average of all the fingerprinting states consistent with the revealed substring.

The intuition is that, on the one hand, ℓ is large enough such that by revealing ℓ bits of k^i Peggy really proves that she knows k^i, while on the other hand the number of non-revealed bits $(d-\ell)$ is large enough to prevent forgeries.

5.2 Substring positions

We set S, ℓ, d such that $S(d-\ell) = d$. For $s \in \{0, \ldots, S-1\}$ we define disjoint subsets $\mathcal{I}(s) \subset [d]$ with $|\mathcal{I}(s)| = d - \ell$,

$$\mathcal{I}(s) \stackrel{\text{def}}{=} \{s(d-\ell), \ldots, (s+1)(d-\ell) - 1\}. \tag{24}$$

The subset $\mathcal{I}(s)$ points at the non-revealed positions in the private key. For convenience we define a 'small' parameter α as $\alpha \stackrel{\text{def}}{=} 1/S$.

5.3 Protocol steps

System setup

Choose message length K, alphabet size S and Hilbert space dimension d, with S dividing d. The parameters ℓ and α follow as $\ell = d - d/S$, $\alpha = 1/S$. Choose an error correcting code $\mathcal{C}$ (over the alphabet $\mathcal{S}$) with codeword size N that can correct symbol error rate θ. Set error tally thresholds $z_{\text{acc}}, z_{\text{rej}} \in \{0, \ldots, N\}$, with $\alpha N \leq z_{\text{acc}} < z_{\text{rej}} \leq 2\theta N$.

Protocol

1. **Distribution of public keys**. Peggy generates private keys $(k^i)_{i=1}^N$, $k^i \in \{0,1\}^d$. For $i \in \{1, \ldots, N\}$ she prepares T copies of the public key $|P_i\rangle \stackrel{\text{def}}{=} |\mu(k^i)\rangle$. Each verifier receives $|P_1\rangle, \ldots, |P_N\rangle$.
2. **Signing**. Peggy announces a message $x \in \mathcal{S}^K$. She encodes x to a codeword $c \in \mathcal{S}^N$ in the code $\mathcal{C}$. She signs each individual symbol of c as follows. To sign $c_i \in \mathcal{S}$ she announces the substring $\kappa_i \stackrel{\text{def}}{=} k^i_{[d]\setminus\mathcal{I}(c_i)}$, i.e. k^i without the positions $\mathcal{I}(c_i)$.
3. **Verification**. Victor receives possibly corrupted data $x' \in \mathcal{S}^K$ and $(\kappa'_i)_{i=1}^N$, $\kappa'_i \in \{0,1\}^\ell$. He performs the following actions. Encode x' to $c' \in \mathcal{S}^N$. For all $i \in [N]$ compute the normalized vector ψ_i as

$$|\psi_i\rangle \propto \sum_{k \in \{0,1\}^d:\ k_{[d]\setminus\mathcal{I}(c'_i)} = \kappa'_i} \Big|\mu(k)\Big\rangle. \tag{25}$$

   For all $i \in [N]$ apply the projective measurement $|\psi_i\rangle\langle\psi_i|$ on $|P_i\rangle$. Let $z \in \{0, \ldots, N\}$ be the tally of '0' outcomes. If $z \leq z_{\text{acc}}$ then the result of the verification is `1-ACC`; if $z \geq z_{\text{rej}}$ then the result is `REJ`. In between, the result is `0-ACC`.

6 Analysis of the proposed scheme

6.1 False reject probability per symbol (G)

We look at the case where a public key $|P_i\rangle$ is unchanged after the Verifier receives it, and Peggy correctly signs. We compute the probability that the verification of one symbol fails.

Lemma 6 *When the projection onto $|\psi\rangle\langle\psi|$ is done in step 3, the probability of a '1' outcome in a qudit is given by*

$$\Pr\Big[\text{projection onto } |\psi\rangle\langle\psi|\Big] = \frac{\ell}{d} = 1 - \alpha. \tag{26}$$

Proof: Without loss of generality we take $c_i = S-1$. Then $\mathcal{I}(c_i)$ equals $\{\ell, \ldots, d-1\}$; the state ψ is computed as

$$\begin{aligned} |\psi\rangle &\propto \sum_{a\in\{0,1\}^{d-\ell}} \Big|\mu(\kappa||a)\Big\rangle \\ &= \sum_{a\in\{0,1\}^{d-\ell}} \frac{1}{\sqrt{d}} \left[\sum_{j=0}^{\ell-1}(-1)^{\kappa_j}|j\rangle + \sum_{j=\ell}^{d-1}(-1)^{a_{j-\ell}}|j\rangle\right] \\ &= \frac{2^{d-\ell}}{\sqrt{d}}\sum_{j=0}^{\ell-1}(-1)^{\kappa_j}|j\rangle. \end{aligned} \tag{27}$$

For general $\mathcal{I}$ and κ we have

$$|\psi(\mathcal{I},\kappa)\rangle = \frac{1}{\sqrt{\ell}}\sum_{j=0}^{\ell-1}(-1)^{\kappa_j}\Big|([d]\setminus\mathcal{I})_j\Big\rangle. \tag{28}$$

The probability of outcome '1' is computed as the square of the following inner product,

$$\langle\mu(k)|\psi(\mathcal{I},k_{\mathcal{I}})\rangle = \frac{1}{\sqrt{d\ell}}\sum_{j'=0}^{d-1}\sum_{j=0}^{\ell-1}(-1)^{k_{j'}}(-1)^{(k_{[d]\setminus\mathcal{I}})_j}\langle j'|([d]\setminus\mathcal{I})_j\rangle = \sqrt{\ell/d}. \tag{29}$$

□

From Lemma 6 we see that the parameter G as introduced in Section 3 is given by

$$G = \alpha. \tag{30}$$

6.2 Forgery probability per symbol

We look at the following attack scenario. The attacker observes a valid signature of symbol s. He owns all T existing public keys. His aim is to create a forged signature for a symbol $t \in \mathcal{S}$, with $t \neq s$. We define

$$J \stackrel{\text{def}}{=} \Pr\Big[\text{forged qudit gives projection } 0\Big] \tag{31}$$

$$\varphi \stackrel{\text{def}}{=} T\frac{\log d}{d}. \tag{32}$$

Lemma 7 *Consider the forgery of one symbol. Let the random variable $A \in \{0,1\}^{\ell-d}$ be the part of k unavailable to the attacker. Let the random variable $W \in \{0,\ldots,\ell-d\}$ be the Hamming distance between A and the attacker's guess for A. The success probability for forging one symbol can be expressed as*

$$1 - J = (1-\alpha)\mathbb{E}_W(1-2\frac{W}{\ell})^2. \tag{33}$$

Proof: Let K be the forged signature, with Hamming distance w w.r.t. the correct key k. We have

$$\begin{aligned} \langle\mu(k)|\psi(\mathcal{I}(t),K)\rangle &= \frac{1}{\sqrt{\ell d}}\sum_{j=0}^{\ell-1}\sum_{j'=0}^{d-1}(-1)^{k_{j'}}(-1)^{K_{\mathcal{I}(t)_j}}\langle j'|j\rangle \\ &= \frac{\#\text{correct} - \#\text{wrong}}{\sqrt{\ell d}} = \frac{\ell-2W}{\sqrt{\ell d}}. \end{aligned} \tag{34}$$

Squaring and taking the expectation over W yields (33). □

Conjecture 1 *Let $f(x,r)$ be defined as*

$$f(x,r) \stackrel{\text{def}}{=} \frac{1}{r} \cdot \frac{\sum_{w=0}^{r} w\binom{x}{w}}{\sum_{w=0}^{r}\binom{x}{w}}, \qquad r \leq \frac{x}{2}. \tag{35}$$

The function $f(x,r)$ is decreasing in r.

Corroboration: We verified this numerically for samples of x up to $x = 2^{28}$. □

Remark: The range of x for which we tested the validity of Conjecture 1 entirely covers the numerical results presented in Section 6.6.

Proposition 1 *For a non-matching position the accept probability can be bounded as*

$$1 - J \leq p_1, \tag{36}$$

where

$$p_1 \stackrel{\text{def}}{=} (1-\alpha)\Big[\Big(1 - \frac{\alpha}{1-\alpha} 2h^{\text{inv}}(1-\frac{\varphi}{\alpha})\Big)^2 + \sqrt{\frac{8}{3}} \cdot \frac{\sqrt{d-\ell}}{\ell}\Big]. \tag{37}$$

Proof: For any distribution P of A (from the attacker's point of view) the probability of successful forgery is maximised by outputting the most likely value of A, i.e. $a_0 = \text{argmax}_a \Pr_{A\sim P}[A = a]$. Then $W = |A \oplus a_0|$. We introduce shorthand notation $p_a = \Pr[A = a]$ and $p_{\max} = \max_a \Pr_{A\sim P}[A = a]$. From the space of distributions P satisfying the constraint $\mathsf{H}_{\min}(A) = -\log p_{\max}$ (with fixed $p_{\max}$) we will determine which P maximizes (33). For $a \neq a_0$ we parametrise $p_a = p_{\max} \sin^2\theta_a \in [0, p_{\max}]$. The Lagrangian for the optimisation is

$$\mathcal{L} = \sum_{a:a\neq a_0} p_{\max}\sin^2\theta_a (1 - 2\frac{|a\oplus a_0|}{\ell})^2 + \lambda[1 - p_{\max} - \sum_{a:a\neq a_0} p_{\max}\sin^2\theta_a], \tag{38}$$

where λ is a constraint multiplier. Setting the derivatives w.r.t. θ_a to zero yields

$$0 = [(1 - 2\frac{|a\oplus a_0|}{\ell})^2 - \lambda]\sin 2\theta_a. \tag{39}$$

Unless λ has a special value, (39) implies $\theta_a = 0 \vee \theta_a = \frac{\pi}{2}$ for all a, i.e. $p_a = 0 \vee p_a = p_{\max}$. Eq.(33) is maximal when P has the following form: strings a with low values of $|a \oplus a_0|$ have probability $p_{\max}$, whereas strings a with high values of $|a \oplus a_0|$ have probability 0. The nonzero probability is concentrated within a 'radius' r around a_0, with $r \leq (d-\ell)/2$.

$$\frac{1}{p_{\max}} = \sum_{w=0}^{r} \binom{d-\ell}{w}. \tag{40}$$

Lemma 1 together with $d - \ell = \alpha d$ and $\log p_{\max}^{-1} = d(\alpha - \varphi)$ yields

$$r \geq (d-\ell) h^{\text{inv}}(1 - \frac{\varphi}{\alpha}). \tag{41}$$

From Lemma 7 we have $1 - J = (1-\alpha)[1 - \frac{4}{\ell}\mathbb{E}_W W + \frac{4}{\ell^2}\mathbb{E}_W W^2]$. Using $W \leq r$ we can write $1 - J \leq (1-\alpha)[(1-2r/\ell)^2 + \frac{4}{\ell}\mathbb{E}_W(r - W)]$. Next, using (41) we get

$$1 - J \leq (1-\alpha)\Big[\Big(1 - \frac{\alpha}{1-\alpha} 2h^{\text{inv}}(1-\frac{\varphi}{\alpha})\Big)^2 + \frac{4r}{\ell}\mathbb{E}_W(1 - \frac{W}{r})\Big]. \tag{42}$$

Finally we have to upper bound $\mathbb{E}_W(1-W/r)$.

$$\frac{1}{r}\mathbb{E}_W W = \frac{1}{r}\cdot\frac{\sum_{w=0}^{r}\binom{d-\ell}{w}w}{\sum_{w=0}^{r}\binom{d-\ell}{w}} \stackrel{\text{def}}{=} f(d-\ell,r). \tag{43}$$

Since f is decreasing in r (Conjecture 1) we have

$$\mathbb{E}_W W \geq r\cdot f(d-\ell,\frac{d-\ell}{2}). \tag{44}$$

Without loss of generality we assume that $d-\ell$ is even and write $d-\ell=2z$.

$$\begin{aligned} f(2z,z) &= \frac{1}{z}\frac{\sum_{w=0}^{z}\binom{2z}{w}w}{\sum_{w=0}^{z}\binom{2z}{w}} \\ &= 2\frac{\sum_{w=0}^{z-1}\binom{2z-1}{w}}{\sum_{w=0}^{z}\binom{2z}{w}} \\ &= 2\frac{\frac{1}{2}\cdot 2^{2z-1}}{\frac{1}{2}[2^{2z}+\binom{2z}{z}]} = \frac{1}{1+\binom{2z}{z}/2^{2z}} \\ &\geq \frac{1}{1+\frac{1}{\sqrt{3z+1}}}. \end{aligned} \tag{45}$$

Thus we have

$$\frac{4}{\ell}\mathbb{E}_W(r-W) \leq \frac{4r}{\ell}\frac{[1+\frac{3}{2}(d-\ell)]^{-1/2}}{1+[1+\frac{3}{2}(d-\ell)]^{-1/2}} < \frac{4r}{\ell}[\tfrac{3}{2}(d-\ell)]^{-1/2}. \tag{46}$$

With $r\leq(d-\ell)/2$ this gives

$$\frac{4}{\ell}\mathbb{E}_W(r-W) < \frac{2\sqrt{2}}{\sqrt{3}}\cdot\frac{\sqrt{d-\ell}}{\ell}. \tag{47}$$

□

Note that Proposition 1 invokes Conjecture 1.

Corollary 2 *The expression p_1 defined in (37) can be lower bounded as*

$$p_1 > 1-3\alpha. \tag{48}$$

Proof. First we write out the square in (37) and neglect most of the positive terms, yielding $p_1 > 1-\alpha-2\alpha\cdot 2h^{\text{inv}}(1-\frac{\varphi}{\alpha})$. Then we use $h^{\text{inv}}(\cdot)\leq\frac{1}{2}$. □

6.3 Setting the parameters; asymptotics

Let $d=2^n$, which means that each qudit can be thought of as n qubits. The φ and $1/d$ are exponentially small in n. For large d one can set $\alpha\ll 1$, $\varphi/\alpha\ll 1$. Then $J\approx 1-p_1$ (37) equals $\alpha+2\alpha\cdot 2h^{\text{inv}}(1-\frac{\varphi}{\alpha})$ minus higher order terms. In contrast, a genuine signature has per-qudit error probability $G=\alpha$.

The scheme must allow the verifier to distinguish between error rate α and J.

Proposition 2 *Consider the scheme proposed in Section 5.3, with parameters* $\theta, N, z_{\rm acc}, z_{\rm rej}$ *set as follows as a function of* α, d, T*,*

$$\theta = \frac{\alpha}{2}(1+\nu) \quad \textit{with } \nu \textit{ constant} > \alpha \tag{49}$$

$$N = \frac{1}{\alpha^3}\left(\frac{\sqrt{3\ln\varepsilon_{\rm c}^{-1}}+\sqrt{1+4\theta}\sqrt{2\ln\varepsilon_{\rm f}^{-1}}}{\frac{1-p_1-\alpha}{\alpha}}\right)^2 \tag{50}$$

$$z_{\rm acc} = N\alpha+\sqrt{N\alpha}\sqrt{3\ln\varepsilon_{\rm c}^{-1}} \tag{51}$$

$$z_{\rm rej} = (1-2\theta)N\alpha+2\theta N(1-p_1) - \sqrt{(1-2\theta)N\alpha+2\theta N(1-p_1)}\sqrt{2\ln\varepsilon_{\rm f}^{-1}} \tag{52}$$

with p_1 *as defined in (37). Given these settings the scheme is correct with error* $\varepsilon_{\rm c}$ *as specified in Def. 1 and* $\varepsilon_{\rm f}$*-secure against forgery as specified in Def. 2.*

Proof. Peggy's signature has error probability α in each individual qudit. The expected tally of errors in the codeword in $N\alpha$. By Lemma 2 and the setting of $z_{\rm acc}$ (51) the probability that the tally exceeds $z_{\rm acc}$ is upper bounded by $\varepsilon_{\rm c}$.

The 'minimal' forgery consists of modifying one symbol in the message x. Let $\tilde{x} \in \mathcal{S}^K$ be the modified message and let $\tilde{c} \in \mathcal{S}^N$ be the corresponding codeword. The Hamming distance between $\tilde{c}$ and c is $2\theta N$. There are $N(1-2\theta)$ symbols that the attacker does not have to modify; in these positions the error rate is α. In the $2\theta N$ positions that the attacker must modify he introduces an error rate J. Hence the expected overall error tally for the forgery is $E = (1-2\theta)N\cdot\alpha+2\theta NJ = N\alpha+N\cdot 2\theta(J-\alpha)$. Eq. (52) has the form $E-\sqrt{E}\sqrt{2\ln\varepsilon_{\rm f}^{-1}}$; substitution into Lemma 2 yields the correct bound on the forgery probability.

We have to enforce $z_{\rm rej} < 2\theta N$. We do this by ensuring that $N\alpha+N\cdot 2\theta(1-p_1-\alpha) < 2\theta N$. This condition can be written as $2\theta(1-[1-p_1-\alpha]) > \alpha$, which is satisfied because of (49) and $p_1 < 1$.

Finally we have to enforce $z_{\rm rej} > z_{\rm acc}$. Setting N as in (50) ensures that this condition is satisfied, as can be verified by a straightforward but tedious computation. □

Proposition 2 depends on Conjecture 1.

Note in (50) that asymptotically N is of order α^{-3}. Setting α to be small has advantages, but these advantages can be exploited only if the signed message is sufficiently long.

6.4 Non-repudiation

Lemma 8 *With the parameter settings given in Proposition 2, our scheme is* ε*-secure against non-repudiation as defined in Def. 3, with* $\varepsilon = T\cdot\max(\varepsilon_{\rm f}^{\nu^2}, \varepsilon_{\rm c}^{\nu^2})\cdot[1+\mathcal{O}(\alpha)]$.

Proof sketch: Malicious Peggy can prepare any state, but all T verifiers receive the same state. Peggy's best chance of causing Rejects as well as 1-Accepts at the different verifiers is to (i) fix the message x before distributing the public key and then (ii) as the 'public key' in each of the N positions prepare a state that is tuned to cause error probability J, with $J = (z_{\rm acc}+z_{\rm rej})/(2N)$. We use Lemma 2 with $\mu \to (z_{\rm acc}+z_{\rm rej})/2$, $\delta \to (z_{\rm rej}-z_{\rm acc})/(z_{\rm acc}+z_{\rm rej})$ to obtain

$$\Pr[Z \le z_{\rm acc}] \le e^{-\frac{1}{4}\frac{(z_{\rm rej}-z_{\rm acc})^2}{z_{\rm acc}+z_{\rm rej}}}, \quad \Pr[Z \ge z_{\rm rej}] \le e^{-\frac{1}{4}\frac{(z_{\rm rej}-z_{\rm acc})^2}{z_{\rm acc}+z_{\rm rej}}}\cdot[1+\mathcal{O}(\alpha)]. \tag{53}$$

From Proposition 2 we get

$$z_{\rm rej} - z_{\rm acc} = \frac{2\theta/\alpha - 1}{1 - p_1 - \alpha}\left(\sqrt{3\ln\varepsilon_{\rm c}^{-1}} + \sqrt{2\ln\varepsilon_{\rm f}^{-1}}\right)^2 [1 + \mathcal{O}(\alpha)] \quad (54)$$

$$z_{\rm rej} + z_{\rm acc} = \frac{2}{\alpha}\left(\frac{\sqrt{3\ln\varepsilon_{\rm c}^{-1}} + \sqrt{2\ln\varepsilon_{\rm f}^{-1}}}{1 - p_1 - \alpha}\right)^2 [1 + \mathcal{O}(\alpha)]. \quad (55)$$

Substitution into (53) yields the same expression for both bounds,

$$\exp[-\frac{\nu^2}{8}\left(\sqrt{3\ln\varepsilon_{\rm c}^{-1}} + \sqrt{2\ln\varepsilon_{\rm f}^{-1}}\right)^2] \leq \max(\varepsilon_{\rm f}^{\nu^2}, \varepsilon_{\rm c}^{\nu^2}). \quad (56)$$

Finally we use (13) in Corollary 1. □

Note that the bound in Lemma 8 is not tight. We present the lemma mainly to show that the repudiation probability is under control. A more complete treatment of non-repudiation, instead of merely a heuristic, is left for future work.

6.5 How many qubits of public key are spent per signed message bit

We show that the size of our public key, taken per signed message bit, can be significantly smaller than in GC01. Expressed in qubits, our public key has size $N \log d$. The length of the message is K nonbinary symbols, which is equivalent to $K \log S$ bits. Asymptotically $K \to N[1 - h(\theta)]$, where $\theta \approx \alpha/2$ if the parameters are set according to Proposition 2. Thus we can write

$$\frac{\#\text{qubits}}{\#\text{bits}} = \frac{N \log d}{K \log S} \approx \frac{\log d}{\log S} \cdot \frac{1}{1 - h(\theta)}. \quad (57)$$

Furthermore, we have the requirement $d - \ell \geq T \log d$, since the number of unknown bits in k must not be smaller than what can be learned from T copies of a d-dimensional qudit. This requirement can be rewritten as $d \geq ST \log d$. Substitution into (57) gives

$$\frac{\#\text{qubits}}{\#\text{bits}} \geq (1 + \frac{\log T + \log\log d}{\log S})\frac{N}{K} \approx (1 + \frac{\log T + \log\log d}{\log S})\frac{1}{1 - h(\theta)}. \quad (58)$$

Compared to the GC01 expenditure (21) our scheme is more efficient by a factor of roughly $\log S$. In theory it is possible to set $S \gg T$ to obtain a public key whose size is just slightly more than one qubit per signed message bit.

6.6 Numerics

We numerically compare our scheme against GC01-with-codewords. We assume an efficient form of GC01 that re-uses unspent quantum states, i.e. if $|P_j^0\rangle$ is measured then $|P_j^1\rangle$ gets relabeled in some way and re-used later.

We fix the number of verifiers T and the error parameter $\varepsilon_{\rm f}$. We focus on three performance indicators: (i) how many qubits are spent per signed message bit, (ii) the gap $J - G$, and (iii) the codeword length N.

Figs. 1 and 2 show plots of the qubit expenditure versus the gap $J - G$ for several combinations of T and α, with parameter settings as in Proposition 2. At small values of α (large alphabet

Table 2. *Comparison between GC01 and our scheme.*

	tunable	qubits/bit	$1-J$	N	gap
GC	d,γ,β	$\frac{\log d}{1-h(\beta)}$	$\frac{1-(1-4\gamma)^2}{2^{d[1-h(\gamma)-\varphi]}}+(1-4\gamma)^2$	$\frac{\ln(1-Q_{\mathrm{R}}^{\mathrm{forgery}})^{-1}}{\beta J^{\mathrm{GC}}}$	J^{GC}
us	d,α,θ	$\frac{\log d/\log\frac{1}{\alpha}}{1-h(\theta)}$	$p_1=(1-\alpha)\cdot$ $\left[\left(1-\frac{\alpha}{1-\alpha}2h^{\mathrm{inv}}(1-\frac{\varphi}{\alpha})\right)^2+\sqrt{\frac{8\alpha}{3d}}\cdot\frac{1}{1-\alpha}\right]$	$\mathcal{O}(\frac{1}{\alpha(J-G)^2})$ (50)	$1-p_1-\alpha$

size S) the public keys are significantly smaller than for GC01. Fig. 3 has the code length N on the horizontal axis. We see that (for the chosen parameter settings) the gain over GC01 sets in when the message length is of the order of magnitude of 8 kilobytes.

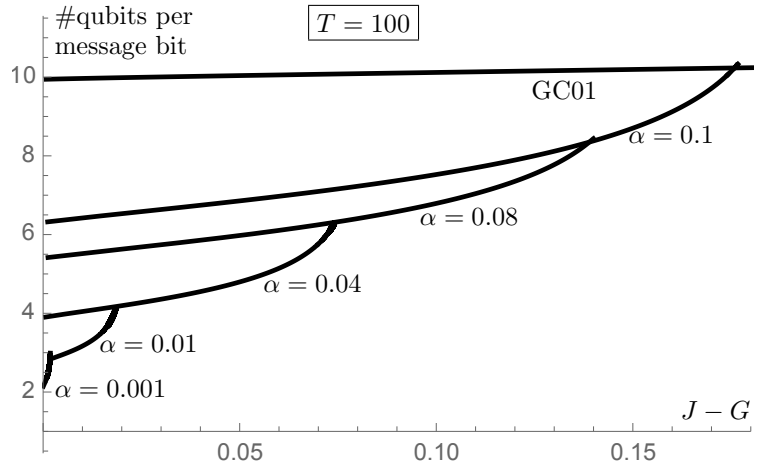

Fig. 1. *Number of spent qubits per signed message bit versus the 'gap' $J-G$, at $T=100$, plotted at various values of α. The almost horizontal curve is the Gottesman-Chuang scheme. In each curve the parameter d is varied. The ranges of d are roughly as follows. At $\alpha=0.001$: $d\in(3\cdot10^6,8\cdot10^8)$; at $\alpha=0.01$: $d\in(3\cdot10^5,8\cdot10^7)$; at $\alpha=0.04$: $d\in(4\cdot10^4,3\cdot10^7)$; at $\alpha=0.08$: $d\in(2\cdot10^4,5\cdot10^6)$; at $\alpha=0.1$: $d\in(2\cdot10^4,7\cdot10^6)$.*

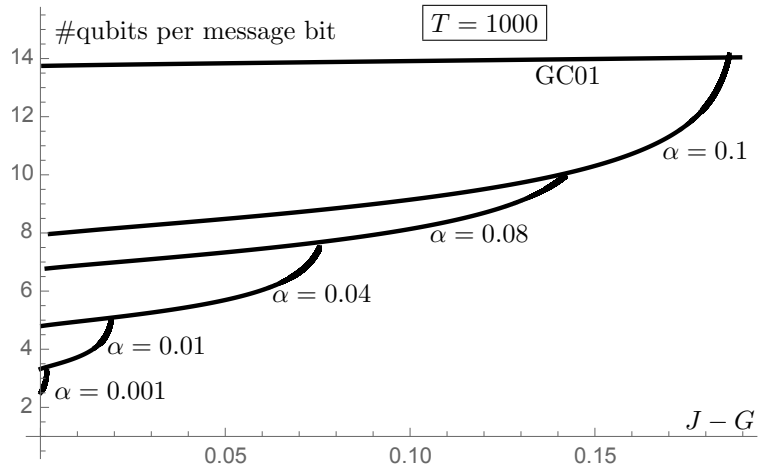

Fig. 2. *Number of spent qubits per signed message bit versus the 'gap' $J - G$, at $T = 1000$, plotted at various values of α. The almost horizontal curve is the Gottesman-Chuang scheme. In each curve the parameter d is varied. The ranges of d are roughly as follows. At $\alpha = 0.001$: $d \in (3 \cdot 10^7, 5 \cdot 10^9)$; at $\alpha = 0.01$: $d \in (2 \cdot 10^6, 4 \cdot 10^9)$; at $\alpha = 0.04$: $d \in (5 \cdot 10^5, 8 \cdot 10^8)$; at $\alpha = 0.08$: $d \in (2 \cdot 10^5, 8 \cdot 10^7)$; at $\alpha = 0.1$: $d \in (2 \cdot 10^5, 2 \cdot 10^9)$.*

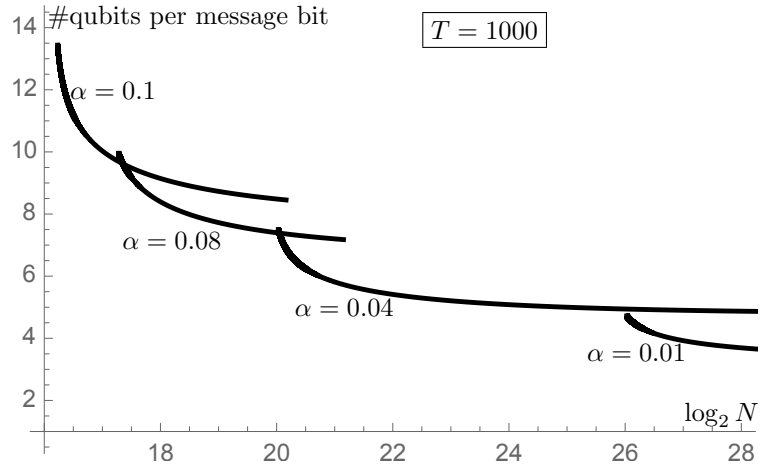

Fig. 3. *Number of spent qubits per signed message bit versus the codeword length N (in bits), for $T = 1000$ and various values of α. $\varepsilon_{\mathrm{f}} = 10^{-12}$; $\varepsilon_{\mathrm{c}} = 10^{-9}$. In each curve the parameter d is varied.*

7 Discussion

It may be possible to improve on the parameter settings given in Proposition 2, and on some of the bounds that we have derived, e.g. by using tighter concentration inequalities.

Our treatment of non-repudiation is less general than that of GC01, who allow Peggy to distribute more general states that pass swap tests. A full treatment would entail determining how large the parameter T (the number of copies) needs to be, as a function of α, in order to ensure that the swap tests sufficiently reduce Peggy's probability of distributing different public keys. Note that Proposition 2 allows a lot of freedom for choosing T; hence we expect that the required T can be accommodated. This analysis is left for future work.

It is rather embarrassing that we do not have an actual proof for Conjecture 1 but only numerical evidence. Fortunately, it is rather straightforward (though time consuming) to verify numerically that the conjecture holds for very large values of x. The graphs plotted in Section 6.6 do not exceed the verified range of x. Hence it is clear that there is a wide parameter regime in which our scheme is advantageous

If Conjecture 1 holds, our scheme asymptotically achieves a public key size of one qubit per message bit. The question naturally arises whether it is possible to go below that value, and if a theoretical lower bound exists. Consider a set of $N = 2^A$ ordinary GC01 public keys labeled $0, \ldots, N-1$, and let the opening of the key with label x represent a signature of the A-bit binary message x. The expended key gets replaced by a new one, while all the other public keys remain in use. Such a scheme spends $(\log d)/A$ qubits per message bit, which can definitely be smaller than 1; however, it requires more complicated synchronisation between the prover and the verifiers than the scheme presented in this paper.

Acknowledgements

We thank Ronald de Wolf, Andreas Hülsing and Aart Blokhuis for useful discussions.

References

1. D. Gottesman and I.L. Chuang. Quantum digital signatures, 2001. https://arxiv.org/abs/ quant-ph/0105032.
2. D. Chaum and S. Roijackers. Unconditionally-secure digital signatures. In *Crypto* 1990, volume 537 of LNCS, pages 206214. Springer-Verlag Berlin Heidelberg, 1991.
3. G. Hanaoka, J. Shikata, Y. Zheng, and H. Imai. Unconditionally secure digital signature schemes admitting transferability. In *Asiacrypt* 2000, volume 1976 of LNCS, pages 130142. Springer Heidelberg, 2000.
4. C.M. Swanson and D.R. Stinson. Unconditionally secure signature schemes revisited. In *Information Theoretic Security (ICITS)* 2011, volume 6673 of LNCS, pages 100116, 2011.
5. L. Lamport. Constructing digital signatures from a one-way function, 1979. Technical Report CSL-98, SRI International.
6. V. Dunjko, P. Wallden, and E. Andersson. Quantum digital signatures without quantum memory. *Phys,Rev.Lett.*, 112:040502, 2014.
7. R.J. Collins, J.D. Ross, V. Dunjko, Wallden P, P.J. Clarke, E. Andersson, J. Jeffers, and G.S. Buller. Realization of quantum digital signatures without the requirement of quantum memory. *Phys.Rev.Lett.*, 113:040502, 2014.
8. R. Amiri and E. Andersson. Unconditionally secure quantum signatures. *Entropy*, 17:56355659, 2015.

9. H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001.
10. D. Gavinsky and T. Ito. Quantum fingerprints that keep secrets. *J. Quantum Inf. Comput.*, 13:583606, 2013.
11. R.B. Ash. Information Theory. Dover publications, 1990.
12. T. Sasaki, Y. Yamamoto, and M. Koashi. Practical quantum key distribution protocol without monitoring signal disturbance. *Nature*, 509:475478, 2014.
13. D. Baron, M.A. Khojastepour, and R.G. Baraniuk. How quickly can we approach channel capacity? In *Asilomar Conf. on Signals, Systems and Computers*, pages 10961100. IEEE, 2004