Why QT™ Unbreakable Matters
(A TED Talk on paper!)

Discussion

Version 1.0 January 2018

All I need is a Miracle…
-Mike Rutherford / Christopher Neil

Author: Mr. Paul McGough, CTO, Qwyit LLC

Abstract

This paper provides a discussion: an abbreviated History of Cryptography – leaving out all the unimportant stuff! As a pre-requisite to this discussion, it's important to first read the QwyitTalk™ Overview, the QwyitTalk™ and Qwyit™ reference guides, the QwyitTalk™ Unbreakable discussion, as well as any and all of our papers and presentations. Or…just read this one!

Contents

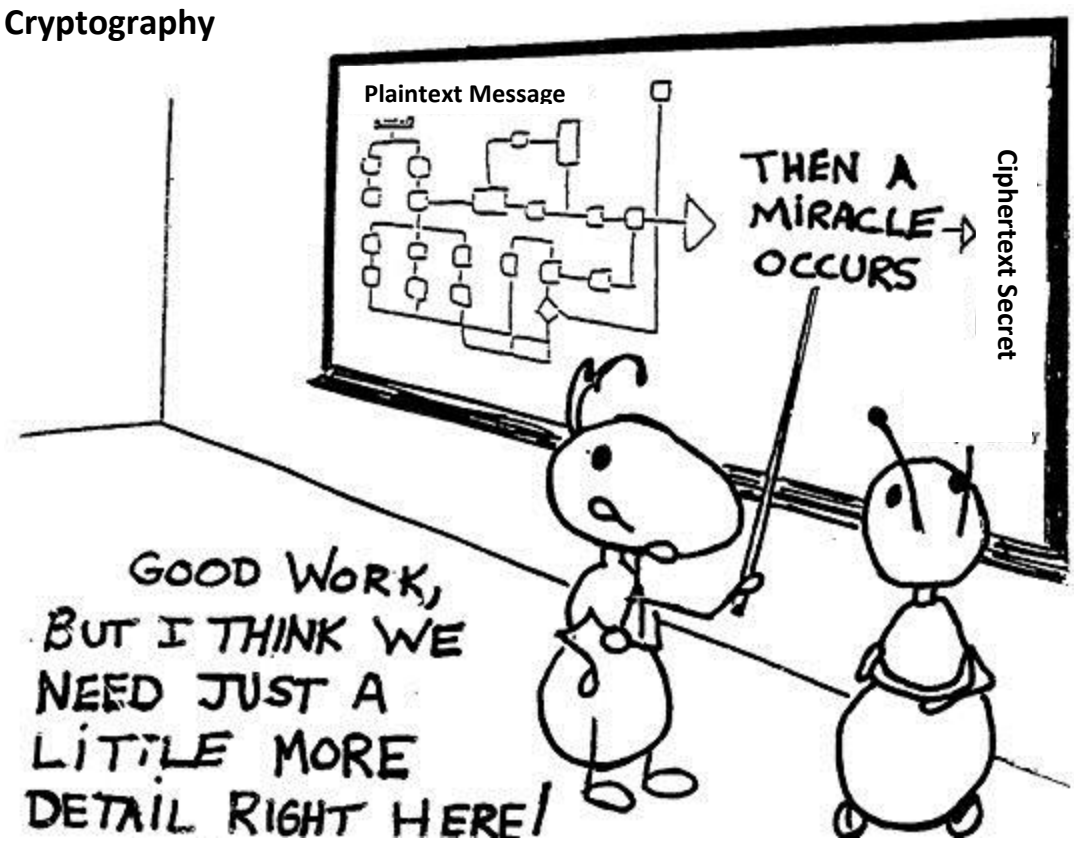
Why QT™ Unbreakable Matters

This document discusses a bit of the history of Cryptography, focusing on how we got here, and where we should go. For more on the Qwyit protocol, and QwyitTalk™ in particular, go to www.qwyit.com.

What's past is prologue – *William Shakespeare*

Introduction

Hello – My name is Paul McGough. I'm an engineer. I'm here today to talk to you about Cryptography. I know what you're thinking…Cryptography?...I must be in the wrong room…I thought this was going to be one of those *FUN* TED talks…and you know what?...I feel exactly the same way! Cryptography – what a *snoozer!!!* And who the heck really knows how that stuff works?! I bet you think it's like this:



There's all this stuff going in…Then A Miracle Occurs…then you have security. And just like it says, we need *just a bit more detail* on exactly what happens. When we take a closer look, it turns out that Cryptography's past isn't leading us to the best possible future. Upon closer examination, I think you'll agree the future should be faster, simpler, more efficient – oh…and Unbreakable.

Keeping a Secret Perfectly

Since this is a 'TED' Talk in words, I need you to ramp up your imagination. It'll be fun – and although Cryptography seems like a daunting (and boring!) subject, it's a lot easier than you think. Let's go:

We've got an audience in an auditorium…imagine the bleachers at your high school gym, with 50 people in one section on the left, then an aisle going down the middle, then another 50 on the right.

Since cryptography is about keeping a secret, the first thing is…*don't tell anyone*!!! But…we're people, so…*we have to tell at least someone*!! Right? So…if all 100 people each had 100 secrets, and they wanted to pick another person out of the group to tell (1 for each different secret), then…well…it would take *hours* for each one to stand up, select someone, then both go out in the hall, share the secret, come back in, sit down…pick the next person…Ah jeez. As that old TV commercial used to say: "There's Got To Be A Better Way!!!" (Is that trademarked?!?! ☺)

Trust: that's the better way. Instead of having to meet each person to keep your trust, what if you trusted a 3rd party, someone, say, standing down at the podium in front of the group (like me!), who each person could come down and leave a sealed message for each of their 100 secrets. Then that ***Trusted 3rd Party*** would ***distribute*** the secret, one at a time, as each of the other 100 came down to leave *their* secrets! OMG!! All it takes is 100 trips down, and *all the secrets are shared correctly!!* I know – this is too simple for *real cryptographers*, but since you might not have known this, this is your introduction to *3rd Party Trust*. So…this podium dweller's name, my name, is ***CA***. It stands for something, and we'll get to that in a minute…but for now, this seems like a perfect solution, doesn't it?

But…(you knew *that* was coming!) What if we did this instead…what if, like a soda vendor, the CA (I'll put a placard around my neck showing you that I'm the CA), I just walk down the aisle starting from the top. As I work my way down, all of the people 'buy soda' – you know, they pass their 'money' (their sealed secrets) down the aisle to the people next to them, who are also 'buying soda's'. When they get to me, the CA (the soda vendor), I pass them back a 'soda' (someone else's sealed secret). By the time I get to the bottom, all done. Not only is this *faster* than all of the people coming down to the podium one at a time (cuz if they all did, I would be an inundated mess!), it's more efficient, easier to do (less envelopes to handle at one time), more flexible (I can move up a row or down one then switch, etc.) – a *whole bunch* of 'better'!!

And:

The Trust is different, isn't it? At Qwyit, we call it 'pass-through trust' instead of 'central trust', like the first example. See – in the central trust, the CA can be bad…and that's…well…*Bad*! In the pass-through trust, the CA, just like all the people in the aisle, can't do anything other than what they're supposed to. No one in the aisle – *even though you trust them not to* – could ever put a secret in their pocket and 'pretend' not to have it! All of you are watching!!! You have to *trust them*, but in reality, *the method holds the trust*. *Same with me, the CA* – I can't just wander off after taking your money and not give you a soda in return – all of the people won't let that happen!!!

So: Are you with me? The first thing to remember about sharing a secret is:

Trust everyone and no one at the same time – and use a method that does that for you. Pass-through Trust.

The Key Distribution Problem Is Just Getting Started

Oh I know what you're thinking…*My secrets need to go to particular people, not just anyone!* Right – so in the example we just used, you could put the name of your intended recipient on each envelop, and for central trust, when you went down to the podium (like going to the Post Office), the CA distributes the secrets as the 'correct' person comes down. And in the pass-through trust, the CA checks each envelop against the folks in the aisle, and gives the secrets to the 'correct' person as well (like a Post Office Mailperson in the truck).

H'mmm…you're thinking the same thing I am, aren't you? Both examples have the *same problem:* How does the CA know who '*correct*' is?!?!

Seems like everyone needs an identifier…a key, a certificate – like the one you get when you're born…

Let's take a look at the current method for handling those keys…or certificates…

The Key Distribution Problem Didn't Go Away

Imagine now, that I'm the CA, and I'm wandering around the gym, ready to Do My Trusted Job: getting your secret envelops to the correct person. Ok?

You there, in the front row on the left side, stand up. Name? "Alice. And I want to send a secret message to Bob – up there in the back corner, right side." Bob stand up.

Ok: Right at this point, since there is a wonderfully unbreakable perfect cryptographic means to send the secret from Alice to Bob, all we need to do is get them both the same, one-time-only secret key.

This is called an OTP, for One-Time Pad – it was invented a century ago and credited to Gilbert Vernam, a Bell Labs Engineer (not a cryptographer, mind you!), and is the only unbreakable cryptographic cipher. It is my job as the CA to perform key distribution, and since I'm the Carrot Top of Cryptography (The *CA* in this case stands for Carrot Top Accessories!) I'll reach into my prop box and throw a Super-Secret OTP Megaphone to Alice…and the same one to Bob. Awesome – they're a little bulky, but they work! Alice, go ahead and tell your secret to Bob. She does…and only Bob can understand her – none of us in the gym can.

Now Bob wants to reply…so I reach into my bag and throw them both another Megaphone – remember, for Gilbert's perfect, unbreakable OTP, they need a new one *every time they talk*. It works again…oh…now you, on the left in the second row, *you say you want to send Bob a secret too?* More Megaphones…Bob, how are you holding up?!?!...He's got this pile of used Megaphones…and he says he wants to talk to 3 more of you on the left side…I'm throwing Megaphones all over the gym!!!…UH-OH!! You see it, right?!?!?

The Key Distribution Problem – we'll call it KDP for short.This ain't gonna work. Can't get keys to everyone, anyone, all of the people, at any of the times they want to talk to whomever they want to talk…OTP keys. Just. Can't. Be. Done.

Lots and *lots* of people, cryptographer people, see this problem…can't solve it…so it has become a '*fact'*: ***we need a different method because we can't solve this problem***. The *Cryptographic Luminaries* of the 1970's came up with asynchronous key systems *as a solution to this KDP.* Asynchronous key systems means that everyone will have two different keys, with two different uses. Since their 'solution' was proposed – and then implemented in the 80's up until now – there hasn't been a single cryptographer in the world that has even tried to solve the KDP.

But – they have this asynchronous key stuff as a solution, so let's see how it works.

First, everyone throw away all those silly Megaphones…and sit down.

The KDP Public Key Solution

Alice, please stand up, and generate a Public/Private Key pair. That's a pair of keys that work in specific ways – we're not going to get into all the hoopla of how, but for now, write your Public Key (the one everyone can see cuz it's Public ☺) on the large placard under your seat. Then write the Private Key, the secret one, in the disappearing ink onto the small placard that has the chain, and put that around your neck. Now hold up the large Public Key above your head for everyone to see.

I, the central-trust CA in this solution, am going to give you a piece of paper…it's called a Certificate. It verifies that you are you, and that the keys you just made (or that I made and gave to you – doesn't matter where they came from), they are *yours and yours alone*! Put the Certificate in your pocket…and get ready to show it to anyone who asks to see it, to verify that you are you, ok? I've got a Certificate just like it that I can show people too. You alright? You've got your Public Key up for everybody to see…you've got a Private Key, that is paired with the Public one, all to yourself…and a Certificate to show anybody who wants to check on you from me that says you're to be believed in – and I have one of those too.

Bob, you ready? Do the same thing…and here's your Certificate from me too. [Please note, everyone: The *Initial Key Distribution* is a different problem – it's not a Cryptographic one – and the method for receiving yours is more-or-less determined by whoever owns the data, the secrets that you're going to share, or the process (device, etc.). Both this current KDP solution, and the one Qwyit proposes, have specific ways to securely get you your first key*s.* Just like you get your first life key, a Birth Certificate, as the basis for your lifelong keys: Driver's License, Passport, etc. OK?!]

Now, Alice, I want you to send Bob a secret message – *right in front of all of these people!* It's gonna be legendary!!

First, you can't really send a message using those Big Ol Public Keys…cuz…well…they're too bulky to deal with, too slow, and they have a specific format that is difficult to work with for routine messages. Take Bob's Public Key, do some fiddling with it, mixing in your Public Key, in those cool Luminary Methods…and when you finish figuring all that out…shoot it over to Bob…if you're having trouble with all the figuring, use that Super Computer in the aisle next to you…you'll need it.

Now…Bob…you do those same cool Luminary Methods – yes, that's what the Super Computer is for in the aisle next to you – Alice used hers too…you make sure it's from Alice…now…both of you, *just to be sure*, and this is…***I'M GOING TO SAY THIS REALLY LOUD CUZ IT'S PART OF THE STUFF THAT GOES WITH THIS SYSTEM BUT NO ONE EVER EVER EVER EVER….EVER….EVER DOES THIS ANYWHERE IN THE WORLD…(***whew!!...that always takes a lot out of me, Old CA!)..***YOU BOTH NEED TO CHECK WITH ME, THE CENTRAL TRUST CA, BEFORE YOU SEND ANYTHING ELSE TO MAKE SURE THE OTHER ONE WASN'T LYING…***and that *none of these good people sitting in between 'Did Something'*…***NO ONE DOES THIS***. But – it's an integral part of the central-trust CA method that is currently used…but …***NO ONE DOES THIS.***

[Have *you* ever stopped at the start of your HTTPS Amazon session, and opened the Certificate to check it…??...and if you did, do you know what you're looking at, what you're looking for??..do you actually go to the Certificate Authority's website to independently verify that what you find in your browser is accurate?!?!...well….??? There. You. Are. Ruminate on that for a second…*an integral aspect of the trust model that is currently being used is 100% completely ignored rendering the trust that is provided to be* ***useless***…yet that is what everyone does…the NSA knows this, criminals know this, hackers know this… everybody but ***you*** knows this…just sayin']

Ok?!...So…now…we're getting there….oh, we're not done yet – because we haven't sent any secret message from Alice to Bob, but…were making progress…slooooow…..slooooooooooooow… progress….

Ok – Now that you've checked w/me (HAHAHA!!! You didn't do that, did you?!)…and used those super computers to compute the stuff I sent to you to make sure everything's up to snuff (which you didn't do either, and that is why you don't do it: ***it would take forever and the system would be useless)***, Alice, Make Up A Secret Key. Got it??...Ok…now – since this key is *sorta, a little bit, but not really the same* as an OTP, but it's the kind of key we can use over and over and over again, you need to send that key to Bob.

Yea, you've only got that Big Public Key, so use that for this message only…bundle it up inside that Public Key (you'll do this with your super computer, using your Private Key) and send it to Bob right in front of all these folks…cuz, just like in our MegaPhone example, he needs to have the exact same key...but he doesn't have it yet, so…go ahead, shoot it over there.

NOW – for this example, we're gonna make it SIMPLE (HAHAHA…has it been *simple so far?!?*), and we'll pretend that you both know to use that key in the same cipher. *In reality,* part of this system, this Public Key Infrastructure (or PKI), includes using lots of different ciphers, so you'd have to include that selection in this message, but…for now, we're going to assume you both know to use the same one. Ok, send it.

Bob, you got it, right? *HEY!* All you other people in the gym…you've got no idea what's going on, do you? So this is *working great so far!!!* Ok, Bob, unwrap that Secret Key that Alice Sent to you. (Yes, use the super computer to do that…)

OK!!!! ***Finally, we're ready to start secretly talking***. Go – Alice to Bob….Bob to Alice…AWESOME!!...back again...and again…yes – *keep using the same key. Again. Again. Again.*

Again. Same Key. Oh…there's 'methods' for twiddling that key…they're kinda cool…like this: Alice, take the last message that Bob sent you, and mix it in w/the key – it doesn't matter, either the ciphertext or the plaintext…yes…just fiddle it in there. Bob, you do the same…so: the key is changing…but…it's just fiddled, not actually different: ***And it isn't 100% OTP Unbreakable***. But…it 'works'…

Cool.

Now – all of you other people in the gym…get out your placards, and make your Public/Private keys…hold them up…connect to the super computer nearest you…and start messaging back n' forth…two round trips before you ever get started talking secretly. Cool.

Everybody alright?...Getting tired?...the rooms kinda full of all these Placards…the super computers are heating up…no one ever checks w/me, the CA (I'm feeling left out!)…it takes quite a while to get ready to message…oh: and there's no relationship between the Public/Private keys on your placards and the Secret Keys that you send, is there? You have to *trust the Public/Private keys and <u>that</u> methodology…and you also have to trust the Secret Key <u>method</u>, don't you?* Two different systems…two sets of problems…two round trips for every message start…too much computing…too much effort…

And NOT 100% OTP Unbreakable.

And this, this PKI, is *the solution? To that simple, original problem: Key Distribution?*

[As an aside, not only is this 'solution' manically complicated and inefficient – the ***main reason it is today's solution is because it is falsely claimed by the Cryptographic Community to have solved the KDP!*** For example, from the University of Rhode Island's Computer Science webpage on Cryptography:

"In asymmetric, or public key, cryptography there is no need for exchanging keys, thus eliminating the key distribution problem." - https://www.cs.uri.edu/cryptography/publickey.htm

Not only have you been falsely told that the KDP is '*solved'*, PKI has <u>*eliminated*</u> the problem!! *Oh really?!?!?!*…IF you were paying just the slightest attention to the above…in the 2nd transmission from Alice to Bob…***<u>PKI SENDS A SECRET KEY…WHICH IS DISTRIBUTING A KEY, ISN'T IT?!?!?!?!?</u>***

Not 'solved'…and certainly not "eliminated"…!!! This technique of imposing something that simply isn't true is called "The Big Lie"…It's chilling in its origins (Hitler), and devastating wherever it becomes a 'fact'…]

I don't know about you…but: (where have we heard this before?!?...) : "There's Got To Be A Better Way!!!"

You cannot escape the responsibility of tomorrow by evading it today. *- Abraham Lincoln*

The Key Distribution Problem IS Solvable

So maybe you've read my papers…maybe you've heard a bit about QwyitTalk™…everybody throw your placards away!! And let's get rid of those super computers too…we don't need 'em. Let's try something else…

Ok…In the same way that you got your Certificate, I'm going to get you your Initial Key – a QwyitTalk™ Key, which is that cool little device under your seat…go ahead…all of you get yours…

It's that little box with the handles on each side, and the window on the front that has the numbers under the glass…DON'T SHOW IT TO ANYONE!!!...ah jeez…you…YES YOU! ..in the back row…BOTH OF YOU…chuck those keys and get the new ones under your seats…I just sent them to you….*HEY!* – did the rest of you notice that?...*Just like at the beginning of this show, when the CA was walking down the aisle, those keys in your hand are secret, important…but…if you screw up, lose one, have one stolen, just like a driver's license, just go get a new one, as the old one will be canceled.*

[We didn't go over getting a new Public Key in the other system…but…you can imagine, right?...some of you remember the old number for Alice, but she's got a new one…and you never ever checked w/me so how would you know?!?...ah jeez…that PKI system's got all kinds of issues, doesn't it?!?]

Ok…now…Alice, send a secret message to Bob. First, flip the handles on your Qwyit Key box and send the numbers you see to me – yea, sure, everybody can see them…now, I've got a big box that has all of your keys – I'm the pass-through CA – and I see that Alice has told me she's gonna message Bob. I put the numbers into my box, and using Alice's starting numbers, flip it, and then I read that indeed, it came from Alice and she wants to talk to Bob.

So, I'm now going to reply with a Alice-Bob-Only Token using a quick flip of my Alice numbers, and also a quick flip of Bob's numbers, to include the same Token for Bob that only he can read, even though I'm replying to Alice. Here ya go, Alice.

Alice do a quick flip, read the Token, and when you use that, it makes you a little Alice-Bob-Only box. Flip those handles to create a Qwyit Secret Key just for You and Bob. This Key is different than the one the PKI guys made…it's only used once, for an exact matching length message part, and it automatically flips the Alice-Bob-Only box to make new keys for every exact matching length message part. These aren't fiddled new keys, they are *mathematically different*. ***They are OTP keys. Unbreakable Keys.***

So now you're all set. Alice…go ahead and start talking secretly to Bob. Your first message will include that part I sent you just for him, so he can verify that you are you, that you got the Token from me, that he can trust you, by trusting me. Now Bob, when you receive that first secret message from Alice, all you've got to do is flip your box that you got from me. That little flip makes the same Alice-Bob-Only box, auto flips its handles to get the starting QwyitTalk Secret Key that Alice created (but *never sent*, as opposed to the PKI method where *real keys are flying all over the place!*), and creates all of the same mathematically different OTP message keys to read Alice's message.

That's It! Alice to me, me to Alice…Alice and Bob talking 100%, unbreakably OTP secure. Just a few handle flips. And now you guys have your own little A-B-Only box, messaging back and forth, flipping the handles each time, creating new *OTP Unbreakable Keys*…

Oh…and after Alice gets that original response from me, the pass-through CA, we both ***independently flip our handles to make a new different number that we both know but is never sent anywhere*** – so all the rest of you have no clue, and no way, to know how we got new numbers.

Oh…and Alice and Bob, go ahead and do the same thing with you're A-B-Only boxes, and flip them independently, without a message between you, whenever the system that you are using to message back and forth says to do it…then you guys can mathematically and separately change *your* Token to create new message keys for the next time…without having to come to me.

Now…ALL OF YOU….ALL AT THE SAME TIME…MESSAGE ANYBODY, EVERYBODY….GO!!!! Flip those handles…I'm right here with you…we're all flipping…messaging each other, everybody, anybody…all with OTP tokens, used only once…making independent QwyitTalk Message Keys that only you and your recipient know (not me, not anybody else!)…all are OTP unbreakable keys…Oh My.

OK!!!...jeez…pretty easy, right?!?...***Key Distribution Problem: TRULY SOLVED***.

Efficient. No super-computers. One quick round to the *actual Trust entity – <u>that never happens in PKI</u>!!* Simple, Fast, Flexible – just flip your handles. Oh my.

<u>Conclusion</u>

Ok – so we didn't show 'the stuff behind the curtains'…the actual methods used to 'figure' in either system. But there's a couple of plain, indisputable facts about that stuff:

- You need 'super computers' for PKI; handle flipping in QT, anyone can do…even little tiny semi-helpless babies in strollers
- There's less 'stuff' in QT – less key stuff, less per-message stuff, less round trip messages
- The Trust is different – simpler, easier to understand, easier to use, *and*…*actually used!*
- Oh…if you look under the hood, the PKI methods rely on theory – they can be broken…QT relies on mathematic fact, it's unbreakable

I'm NOT a cryptographer…because all of the cryptography we ever need was done a century ago: OTP's are unbreakable. ***It's all about distributing those OTP keys, every time, for every use, to anyone who wants to message securely. That is an engineering problem…Those Ants In The Diagram!***

Gilbert Vernam was an engineer. Me too.

QwyitTalk™ is the 'Miracle'…it's a straightforward engineering solution to an engineering problem…and a pretty simple one at that: just a bit of unsolvable math.

Now you know and understand how Cryptography in the Digital Age currently works…and how it *should work* for a better future!

Thank you!