Not Before Time v2.4

Not Before Time is a public broadcast facility built on existing technology. Not Before Time gives three strong guarantees that particular information:

1. ***will not be readable before a certain future time***
2. ***was not created before a certain past date and time***
3. ***was not electronically signed before a certain time***

These support human rights, democracy and business.

The novel features of Not Before Time are:

1. Time locking information is the opposite of the modern and stressful trend for all information to be available instantly, everywhere.
2. The three guarantees work for ordinary users using their everyday software on their existing devices.
3. The mechanism and its novel new applications as easily explained to non-technical people, which is essential for a system to be trusted.

Version 2.4 1st September 2021

Dan Shearer

Table of Contents

Introduction and Overview

Not Before Time provides a new certainty to the age-old question "*What did they know, and when did they know it*?"

Sometimes new certainties arise in human history:

> **Reliable timekeeping**. The first portable clocks were developed to help ships navigate. In the centuries since society has been defined by measured time - the neutral certainty of ticks and tocks. In the 21st century the right of an individual to access accurate time is assumed, and legally it is included in other fundamental rights which are explicitly guaranteed.
>
> **GPS**. In recent years, always-on knowledge of latitude and longitude has also redefined society, although GPS raises uncomfortable questions as well as opportunities. Location knowledge about individuals, machines and the urban environment is increasingly important. Access to GPS is a fundamental individual right protected within the European Union (GPS is always taken to mean GPS and GLONASS together.)
>
> **Radiocarbon Dating**. The certainty of C14 dating made history jump into focus, and dispelled many myths and legends. For the first time it was possible to know when key events happened (such as ice age transitions) and what order they happened in. It was possible to place objects and people in their correct temporal context.
>
> **Evolution**. With the certain knowledge that evolution drives all life it has been possible to begin decoding the petabytes of instructions, historical records and potential encoded in the DNA and the genes built on them. We know we can know. That is a fundamental certainty.
>
> **NBT** provides another new kind of certainty. Anyone can connect physical or electronic information with time in ways that are extremely difficult to subvert.

NBT is a clock that works for information. Like GPS, NBT is a public broadcast technology that enables new creative opportunities for an unlimited number of users. Like GPS, NBT cannot be limited to people of particular views, nationality, employer or any other difference. NBT removes the ultimate choice about the publishing of information from internet and media companies, and gives it back to authors and publishers. NBT is the opposite of chat with disappearing messages, instead, it is a constant stream of messages from the past appearing in the present.

Technically, NBT contains no surprises. NBT is a mildly different view on thirty year-old technology that is already deployed to billions of users worldwide. There are no new principles or fundamental inventions, only the way they are put together.

NBT resists obsolescence. NBT is intended to be effective for decades and longer despite the fact that decryption techniques get faster and better over time. NBT incorporates state of the art of quantum-resistent encryption methods despite also using thirty year-old encryption principles. NBT includes an upgrade method should this be needed. Where the NBT future date is relatively close (say just a handful of years from today, in 2021) the calculable risk of NBT being compromised by the cryptography chosen is lower than that of many other widely-used cryptographic systems.

NBT has immediate and scaleable applicability to finance, unlike cryptocurrencies. NBT's application to financial transactions of all kinds including cryptocurrencies is not discussed in this document. Guarantees of the contents of a time-encrypted finance document has such obvious benefits that it requires standalone thought. This is just one of the many applications for NBT.

Examples of Linking Information to Time

- **Embargo.** Secretive Ltd, wishes to make a massive announcement worldwide. An embargoed press release secured by NBT is circulated to press in all timezones, behind all manner of firewalls and on individual devices which can have moved anywhere. Exposure is maximised because it is already in everyone's hands, and yet the embargo can't be broken.
- **Sealed bids.** Smart Ltd wishes to build a new office. Commercially sensitive bids are received from many construction companies, with NBT being a requirement for all bid submissions, and bids closing on July 17th 2022. The bids cannot be read before the closing date by anyone, not even the construction companies that sent them. Nobody can accuse Smart Ltd of giving secret pricing information from one bidder to another.
- **Proof of earliest possible creation.** A video is encrypted with a secret password only known to Jemimah, and signed with NBT. When Jemimah gives the secret password to someone and they decrypt the video, they know for sure that Jemimah encrypted the video on or after 10am on 11th April 2019, not before then. Nobody can claim they received the NBT-protected video and the secret password in, say, 2017.
- **Physical proof of earliest possible creation.** A controversial paper book or poster is published with a QR code on the front cover. That QR code contains a large, unique and random number published under NBT on 18th September 2020. Therefore, NBT has set the earliest possible publication date for the book or poster.
- **Email Decryption After Receipt.** Using standard Google Email, an encrypted message is received with the subject "This email will be readable from midnight, 25th December 2021", which is in three days' time. Three days later, the email turns to plain text using Google's standard tools for encrypted messages.
- **Unencrypted Email With Proof of Earliest Possible Creation.** A standard Yahoo Email inbox receives a message with the subject "This digitally signed email contains meeting notes that were not sent before 14th October 2021". Despite the fact that nothing is known about the sender's email system, and that email is notoriously easy to fake, NBT proves beyond doubt that the statement is true. For example, a court of law could not accept a claim that the notes were sent on the 10th October 2021, because there is no reasonable doubt.
- **Combined Creation Proof and NBT.** Using standard Google Email, a plain text message is received with the subject "The attached encrypted PDF will be readable from midnight, 25th December 2021, and has on the first page a number which only existed on 1st January 2020." NBT ensures we can be certain that the PDF was not made before January 2020 and will not be read earlier than December 2021. This is less complicated than it sounds, and could feasibly become a part of how business is done.

- **Getting Snowden to Iceland.** Edward Snowden wishes to publish gigabytes of information. He uses NBT to secure a file which is then bittorrented anonymously. Some months later, after he is comfortably settled in Iceland, the world is amazed when the NBT timelock ticks over.

Technical Principle of Operation

This section requires an understanding of the principles of Public Key Cryptography, for example, as described in https://en.wikipedia.org/wiki/Public-key_cryptography .

There are two phases to establishing a Not Before Time network:

1. Publish a static list of public keys far and wide, with one key for every time interval, for example, every half hour. This means there will be a public key corresponding to 0930 in the NBT timezone on the 3rd February in the year 2043. If the system is intended to last just a modest 25 years then that is 350 000 half-hourly keys, requiring about one hundredth of a second to search on a relatively slow computing device.

2. Broadcast each private key corresponding to the public key at the relevant time. This new private key adds to the growing list of all private keys as time passes, allowing everyone everywhere to unlock information. The guarantees are still kept regardless of any local propagation delays, because it is Not *Before* Time rather than Not *At* Time.

The only significant problem to solve is how to keep the list of private keys secret in a trustable manner, and consideration is given to that later in this document. Everything else is straightforward software engineering.

This is an example specification for how NBT can be implemented using everyday technologies.

Specification of the Time Format: RFC 3339 describes standard time formats such as can be derived from standard NTP servers including ones fed by atomic clock sources.

Preparation of the Lists of Public and Private Keys: This is a handful of lines of code run on a computer with excellent sources of entropy, calling GPG to generate public/private key pairs for each time interval. The time interval expressed in the selected time format is both the passphrase for the private key and also the name of the public/private key pair. After validation, this yields two lists. The process of keeping the private key list secret in a trustable manner is the key challenge for NBT.

Loading the Private Keys into the NBT Server and Starting the NBT Server: The NBT server has an independent and reliable time source (eg multiple atomic clocks checked with GPS). The list of private keys corresponding to the list of public keys is stored within the NBT Server and *only* within that server. The NBT Server then publishes each private key once the time it corresponds to has passed. The NBT Server may be a little late, but never early. The method of publication can include DNS, RestAPI, web page, and more.

Preparation of the Message by an Individual User: A text message is encrypted (eg on a personal computer, with nobody else able to see) using the public key corresponding to the “Not Before” time. This means that the public key has to be fetchable in an automated way, or pre-loaded on the device beforehand, since storage requirements and access times will rarely be an issue. Proof of concept code demonstrates that there are multiple obvious intercept points for users, including File Save/As dialog boxes asking for a future time to lock the document. It is well-known that public key cryptography in email is possible but painful, but the painful part is about key handling. In this case the key handling is all very well known sources and destinations.

Distribution of the Encrypted Message by an Individual User: The encrypted message is then distributed using whatever is appropriate. It might be sent as a private email to half a dozen people. It might be put on a public webpage. It might be sent to a server which will wait until the target time has passed, and then distribute it once it has been decrypted.

Decryption of the Message by the Recipients or People Generally: The encrypted message can be manually decrypted using any PGP-compatible program such as GnuPG once the “Not Before” time has passed. Automated services such as email mailstores and content management systems can watch the NBT keyserver for the appropriate private key release and then decrypt the message immediately. Proof of concept investigations have shown that building PGP into end-to-end encrypted chat systems is not difficult, especially since cryptocurrency support is becoming more popular in chat.

Technical Description of Guarantees

Putting the three guarantees in the language of Computer Science, NBT gives strong cryptographic certainty to the users wishing to share information so that they can:

- choose a date and time before which the information is unreadable, using accepted algorithms and implementations;
- prove that the information was not electronically signed and sealed before a certain date/time (which is new because electronic signatures do not currently incorporate a timestamp in a universally trusted manner);
- prove conclusively that information was not created before a certain time (which is new because electronic signatures do not make any statement about the original source of the material they are authenticating)

I think of these in shorthand as:

1. No Early Reading
2. No Late Denials
3. No Digital or Physical-world Backdating

Technical Trust & The Private Key Server

The NBT private key server is in principle a single point of failure, either through technical failure or security breach. An attacker with a secret copy of all private keys would be in a very strong position, because if all users trust the system this attacker can quietly monitor for NBT-locked data and decrypt it for their own nefarious purposes.

NBT lends itself to theoretical trust analysis in well-studied fields of computer science, and also a class of grandstanding and spectacular physical solutions.

All but two of the solutions require new thinking.

The fundamental idea is that no one group of implementers can be trusted for NBT. For example if Russian computer scientists deploy the system then Americans will claim that the FSB is in possession of all secret keys, and if the Swiss government sponsors a neutral NBT server then the Chinese government will claim that the CIA stole the secret list. If Seiko or some other timekeeping company decide this is their future then Amazon will issue legal threats to stop them and so on.

This doesn't just stop with initial creation. Even if the NBT server is set up so that it automatically creates the two lists without human intervention and manages them from then on, and even if the world can agree on the entity to do that work, someone else will claim that the code involved was compromised (or the computer doing the work was compromised before it was locked down) and so on. And even if the notional NBT server was created securely, can we trust where it is housed? A stratum zero timesource is exceptionally well-protected, and we can install alarms, failsafes and more and yet we will know that if enough money, intellect and political will is focussed on the problem then it can be cracked. Perhaps a deceased person is suspected of NBT-locking kompromat on a powerful person, or nuclear secrets, before they died. While in fact it might be a letter to their children, there would be significant incentive to attack the system.

Using traditional internet approaches, the only solution I have yet found using is to avoid any single point of failure by using fully decentralised technology and zero-knowledge algorithms where mutually untrusted parties each have a piece of the list. This assumes a trusted one-off list generation, but having already engaged mutually mistrustful parties this can be be arrange. On the whole it is relatively difficult problem but some parts of it are known to be solvable, including in some published and peer-reviewed code used at scale.

There are three cases where the problem becomes easy to solve:

1. For an initial commercial live demonstrator. For some of the commercial applications we have defined for the legal and construction industries, the level of security currently used is sufficiently low that a special-purpose NBT would be an improvement. It also would not be a global high-profile target.
2. For a time-limited global demonstrator. This could be the same as the commercial live demonstrator, only that public keys only go out say two years. Again, this is unlikely to become a high-profile target, so we don't have to take the most extreme care to be trustable.
3. Launching an NBT server into space. This is cheaper and more practical than it has ever been before. Mutually-mistrustful protocols are still required for the preparation phase, but

there is no practical security issue after a successful launch. The beauty of NBT is that propagation delays aren't vital, which saves a lot of orbital trajectory calculations. This is the very definition of a single point of failure, so multiple launches with the same or different secret key lists could be arranged. If using more than one satellite, then the private key list can be transferred between satellites from the oldest-launched to the newest using a line-of-sight technology such as a laser. There are many ways for a satellite solution to fail, but not many ways for it to be compromised once off the launchpad. Particularly since none of the potential attackers know that NBT is going to be a success and should therefore be a high-value target.

Steps to a Demonstration Project

The result will be a fully functional prototype by following these steps:

- Create an NBT key server distribution service via protocols such as Rest, DNS, and possibly a private instance of MIT Keyserver. (DNS has fields that can be used and/or highjacked for NBT purposes.)
- Create an NBT client as a web service, which receives unencrypted files and returns them encrypted to an NBT future date, and also receives encrypted files and either decrypts them or states when they can be decrypted. This can be just a wrapper on GnuPG.
- Create a virtual printer device for the Linux operating system, that functions in much the same way as "Print to PDF", except that it prints to an NBT-locked PDF.
- Create a small modification to LibreOffice, where File/Export As PDF OR File Save/As has a new field for "Not Readable Before This Date", fetches the appropriate public key from the NBT keyserver, and encrypts the generated file. Key libreoffice developers estimate 200 lines of code need changing, and some scripting experiments have been done.
- Create a modified PDF reader OR a corresponding modification to File/Open in LibreOffice where if an NBT header is detected then the NBT server is polled to see if there is a private key released. The poll is not necessary if the local time is regarded as accurate.

For production use, there are equivalents to all of the above for most common commercial and open source systems. Intercept vectors such as printer drivers, File/Save and File/Open dialogs are routinely hooked. Closed source SaaS products from Office 365 to Salesforce have large aftermarket communities with addon hooks. Android and iOS development contractors assure me there are multiple ways to solve this without inventing a new class of mobile solutions.

Enabling Existing Email Infrastructure

One early next step should be emails. Currently almost nobody has an everyday motivation to encrypt emails, despite the facts that it improves security and privacy. Even if key management was not a problem, why would anyone bother? Security and privacy are not major motivators for most people including technical people. The GDPR and ePrivacy laws are not exciting motivators for security, compliance never is.

However, emails that are only visible after a certain time... That's instantly graspable by most people. NBT could be a reason for everyone to use the encrypted mail facilities they already have, since the key management problem shrinks to zero in the case of NBT. One little plugin, for many email users, and they will have access to cool NBT facilities.

Of course even the possession of an encrypted email conveys information. I have developed a spec for sending emails Not Before Time, an email server that sits on encrypted emails and chat messages until they can be read. But that is a separate story and a separate online service.

GPS As a Historical and Commercial Model

The broadcast facility is base infrastructure without a direct commercial model, in the same way as GPS and GLONASS which combined are in the location chipsets used in virtually all mobile devices, and increasingly other satellite location systems too. It does not make sense to try to limit access to the NBT broadcast facility, even through that would be possible by means of adding a second layer of keys. The American GPS was limited in accuracy for years to the US military or those that cracked the encryption code in GPS. Then it was realised that the applications being built on top of GPS regardless of these restrictions were of immense commercial value (and perhaps as an afterthought, of immense value to the preservation of human life and improvement of the human condition...)

For NBT, the immediate monetisable value is in applications on top of this base infrastructure. The human rights use case cannot be shut down without also shutting down commercial activity, so that from launch we have a facility which is a high-value target.

ENDS