Working Towards Digital Archive Transparency

It is time to add cryptographic timestamps to all the things

---

archive.rip

July 2nd, 2025

Working towards Digital Archive Transparency

1. Introduction
2. What is the problem?
3. How can we fix this?
4. What about transparency logs?
5. Where I'm at? (a work in progress)

1. Introduction

What, When, Where

This applies to:

- online digital archives (The Internet Archive, ...)

What, When, Where

This applies to:

- online digital archives (The Internet Archive, ...)
- other entities "distributing documents" (in a wide sense)

What, When, Where

This applies to:

- online digital archives (The Internet Archive, ...)
- other entities "distributing documents" (in a wide sense)

We will not focus on *"how"* these documents are distributed:

What, When, Where

This applies to:

- online digital archives (The Internet Archive, ...)
- other entities "distributing documents" (in a wide sense)

We will not focus on *"how"* these documents are distributed:

We want to authenticate their
"when" & "where"

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
  +You check the domain name (e.g nist.gov not nist.rip)

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
  +You check the domain name (e.g nist.gov not nist.rip)
  +You check that your connection is secure

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
  +You check the domain name (e.g nist.gov not nist.rip)
  +You check that your connection is secure
- The NIST website seems to be the document's **primary origin**

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
  +You check the domain name (e.g nist.gov not nist.rip)
  +You check that your connection is secure
- The NIST website seems to be the document's **primary origin**
  The website & document metadata say "published in 2015"

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website seems to be the document's **primary origin**

trusting metadata? in this economy?

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website seems to be the document's **primary origin**

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website **today** seems to be the document's **primary origin**

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website **today** seems to be the document's **primary origin**

You learned two things from that informal protocol:

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website **today** seems to be the document's **primary origin**

You learned two things from that informal protocol:

- The *"where"*,

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website **today** seems to be the document's **primary origin**

You learned two things from that informal protocol:

- The *"where"*, some server you *securely connected to*

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website **today** seems to be the document's **primary origin**

You learned two things from that informal protocol:

- The *"where"*, some server you *securely connected to*
- The *"when"*,

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website **today** seems to be the document's **primary origin**

You learned two things from that informal protocol:

- The *"where"*, some server you *securely connected to*
- The *"when"*, **today**,

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website **today** seems to be the document's **primary origin**

You learned two things from that informal protocol:

- The *"where"*, some server you *securely connected to*
- The *"when"*, **today**, you have learned nothing about the past

Roleplay (as a cryptographer)

Why do we need to *establish trust* into digital archives?

- Imagine you are a cryptographer named Alice 🧙‍♀️
- Bob gives you a NIST paper he downloaded **ten years ago**
- You search for it online, to "verify" the document is *legitimate*
- The NIST website **today** seems to be the document's **primary origin**

You learned two things from that informal protocol:

- The *"where"*, some server you *securely connected to*
- The *"when"*, **today**, you have learned nothing about the past

You were not there ten years ago!

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today**

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗
  Maybe you're talking to a bad actor?

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗
  ~~Maybe you're talking to a bad actor?~~
  Some legitimate modification not listed explicitly?

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗
  ~~Maybe you're talking to a bad actor?~~
  Some legitimate modification not listed explicitly?
  Some security issue, was modified 5 years ago, no one noticed?

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗
  ~~Maybe you're talking to a bad actor?~~
  Some legitimate modification not listed explicitly?
  Some security issue, was modified 5 years ago, no one noticed?
  Some new management, a new board with new policies?

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗
  ~~Maybe you're talking to a bad actor?~~
  Some legitimate modification not listed explicitly?
  Some security issue, was modified 5 years ago, no one noticed?
  Some new management, a new board with new policies?
  You just don't know what could have happened!

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗

There is no way around it:

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗

There is no way around it:

You have to trust someone

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗

There is no way around it:

You have to trust someone (someone in the past, ten years ago)

Informal Primary Origin Authentication

This informal protocol is very limited in its results:

- I saw that document **today** on the (official) website ✓
- Nothing *proves to you* that it was the same *in the past* ✗

There is no way around it:

You have to trust someone (someone in the past, ten years ago)

What about people *ten years in the future*? [1]

[1] note that to these future people, we are that "someone in the past" :)

Is this really an issue?

no problem if document stored on me computer

2. What is the problem?

Relying on an Informal Chain of Trust

The informal approach is flawed, because we have monkey brains

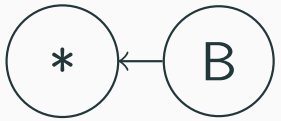
Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**

Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**

(primary origin)*



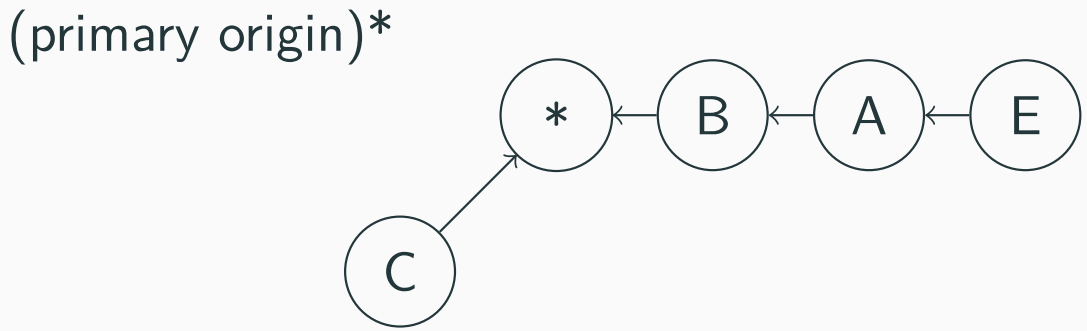
Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**

(primary origin)*

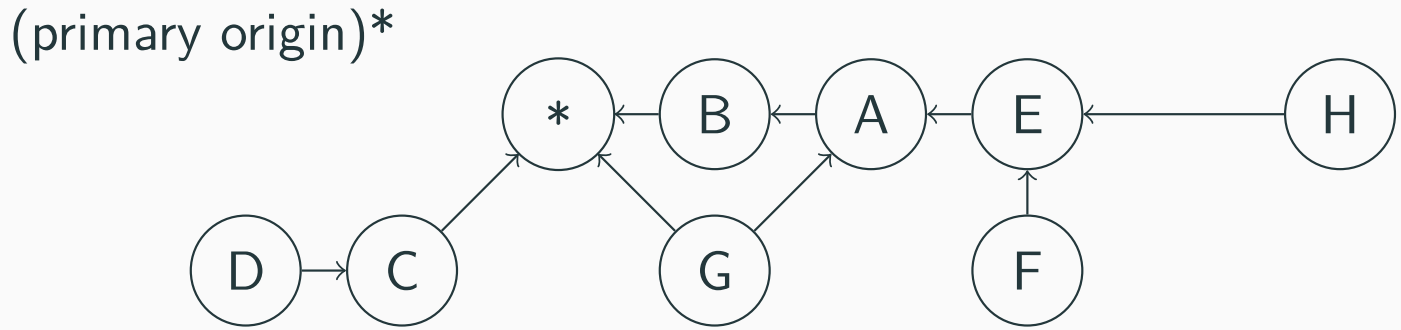
Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**



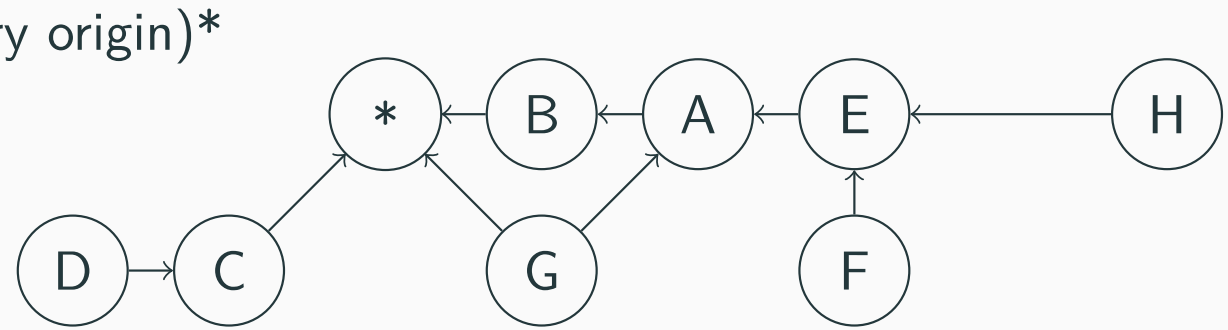
Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**



Relying on an Informal Chain of Trust

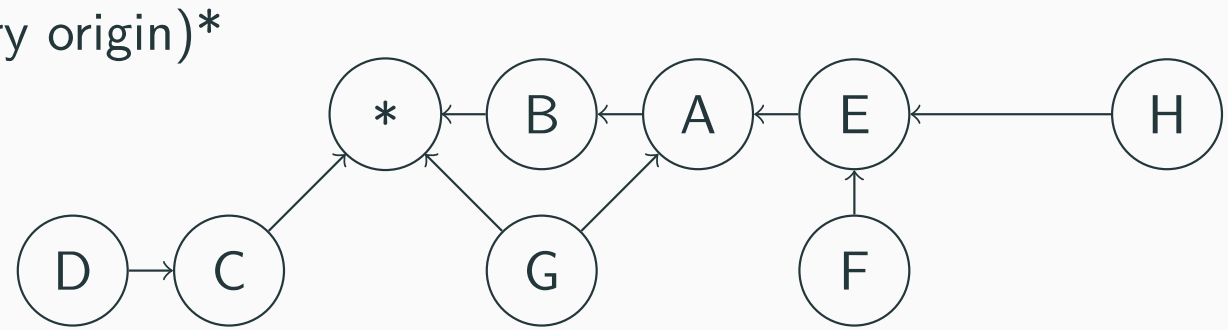
The informal approach is flawed, because it is **online & interactive**



- A **chain of trust** relying on its weakest link :(

Relying on an Informal Chain of Trust

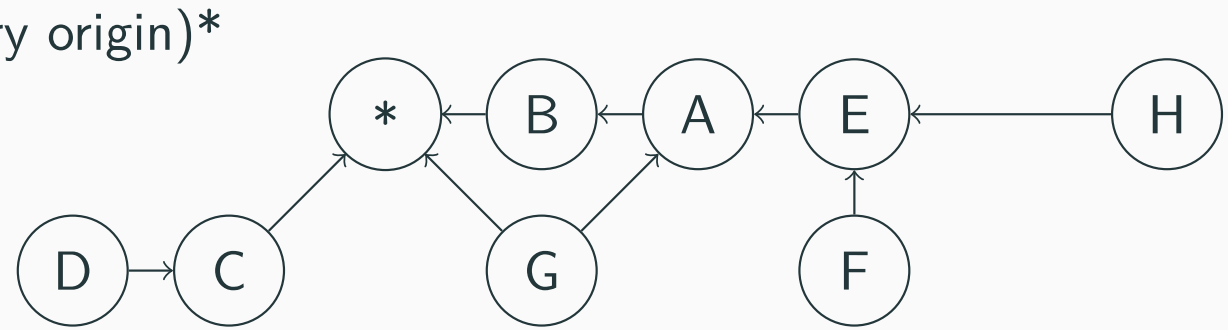
The informal approach is flawed, because it is **online & interactive**



- A **chain of trust** relying on its weakest link :(
- Funding?

Relying on an Informal Chain of Trust

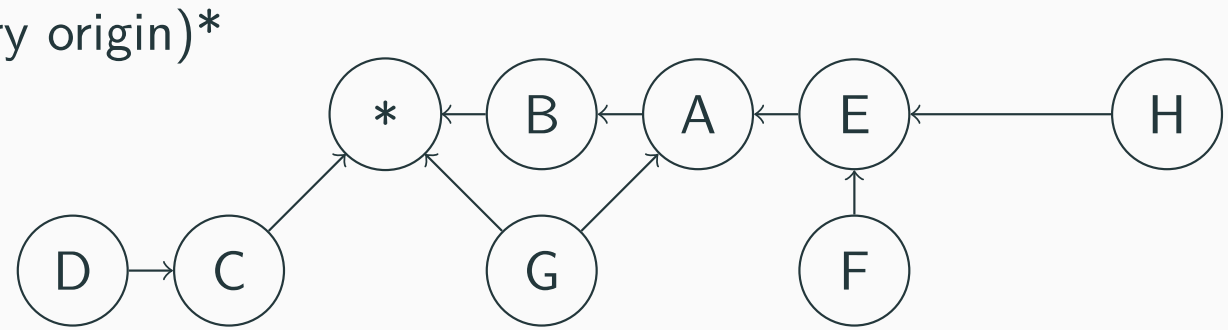
The informal approach is flawed, because it is **online & interactive**



- A **chain of trust** relying on its weakest link :(
- Funding? Oceanic cables?

Relying on an Informal Chain of Trust

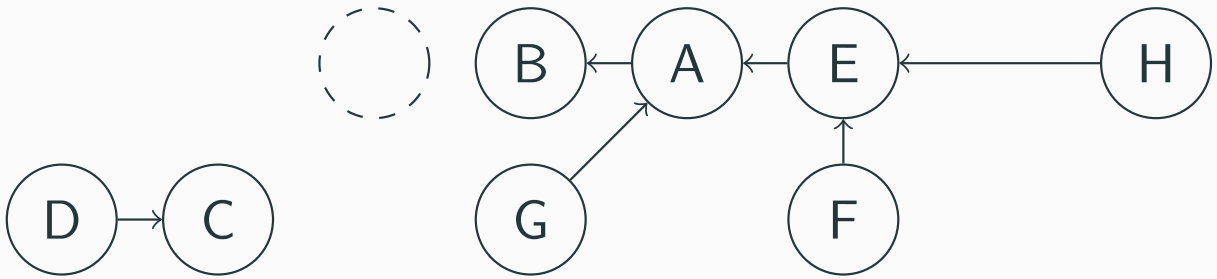
The informal approach is flawed, because it is **online & interactive**



- A **chain of trust** relying on its weakest link :(
- Funding? Oceanic cables? Neglect?

Relying on an Informal Chain of Trust

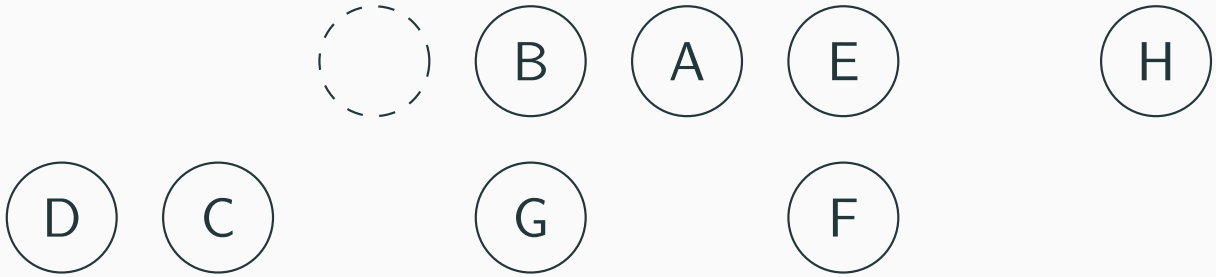
The informal approach is flawed, because it is **online & interactive**



- A **chain of trust** relying on its weakest link :(
- Sooner or later, you will loose your **online source of trust** :(

Relying on an Informal Chain of Trust

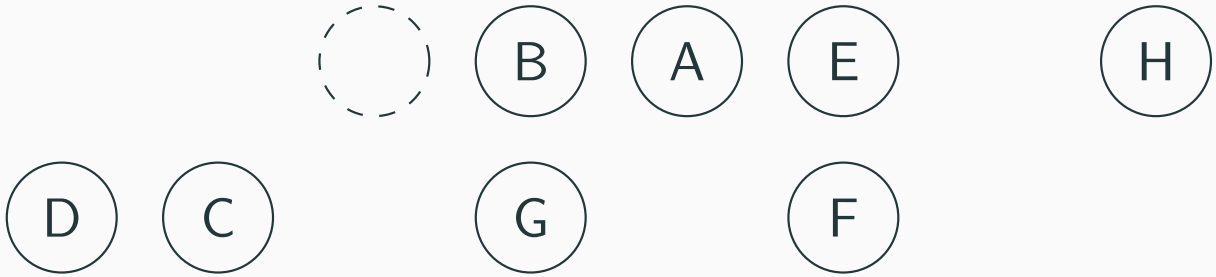
The informal approach is flawed, because it is **online & interactive**



- An *informal* **chain of trust** relying on its weakest link :(
- Sooner or later, you will loose your **online source of trust** :(

Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**

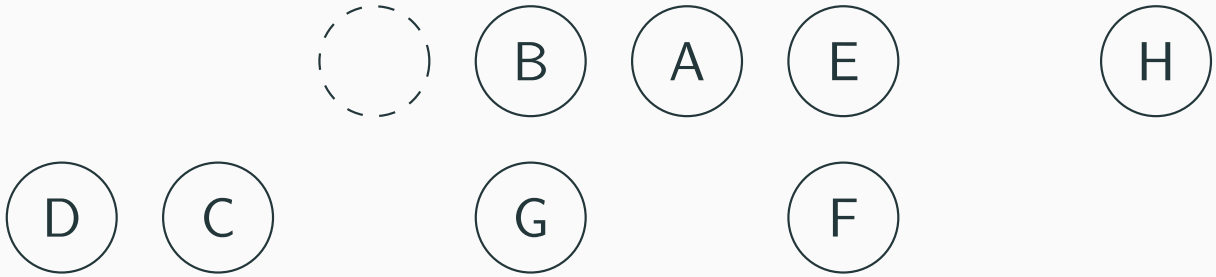


- An *informal* **chain of trust** relying on its weakest link :(
- Sooner or later, you will loose your **online source of trust** :(

What if a bad actor tampers with our digital records?

Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**

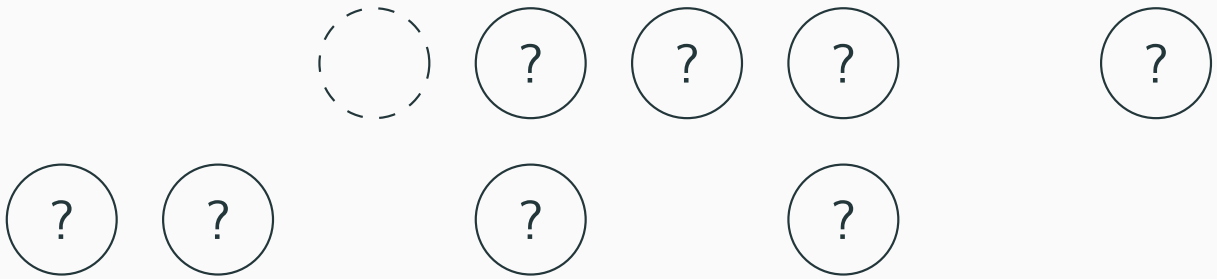


- An *informal* **chain of trust** relying on its weakest link :(
- Sooner or later, you will loose your **online source of trust** :(

What if a bad actor tampers with our digital records?
(or merely convincingly claims it has)

Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**

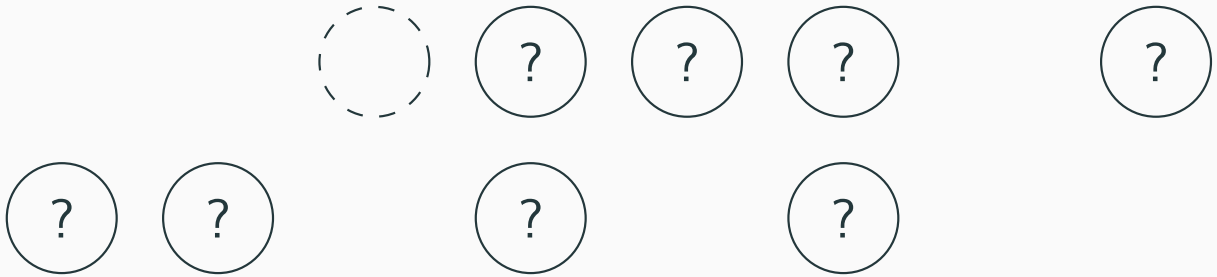


- An *informal* **chain of trust** relying on its weakest link :(
- Sooner or later, you will loose your **online source of trust** :(

What if a bad actor tampers with our digital records?
(or merely convincingly claims it has)

Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**



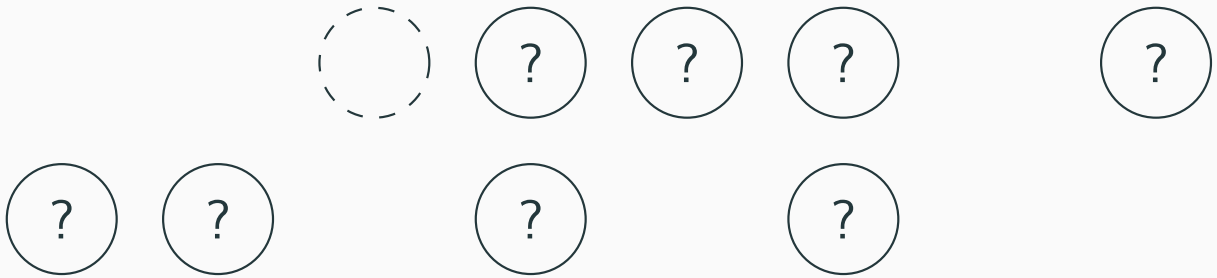
- An *informal* **chain of trust** relying on its weakest link :(
- Sooner or later, you will loose your **online source of trust** :(

What if a bad actor tampers with our digital records?
(or merely convincingly claims it has)

We have no way to "revert back" to a "trusted past" 😟

Relying on an Informal Chain of Trust

The informal approach is flawed, because it is **online & interactive**



- An *informal* **chain of trust** relying on its weakest link :(
- Sooner or later, you will loose your **online source of trust** :(

What if a bad actor tampers with our digital records?
(or merely convincingly claims it has)

The digital past is **indistinguishable from backdated data**

”Just go to The Internet Archive”

There exist **online digital archives** with snapshots of the past 🎉

"Just go to The Internet Archive"

There exist **online digital archives** with snapshots of the past 🎉

- They become the new weakest link of the chain of trust :(

”Just go to The Internet Archive”

There exist **online digital archives** with snapshots of the past 🎉

- They become the new weakest link of the chain of trust :(
- We can not expect perfect operational security for decades :(

”Just go to The Internet Archive”

There exist **online digital archives** with snapshots of the past 🎉

- They become the new weakest link of the chain of trust :(
- We can not expect perfect operational security for decades :(
- They can not afford **becoming a target for bad actors** :((

"Just go to The Internet Archive"

There exist **online digital archives** with snapshots of the past 🎉

- They become the new weakest link of the chain of trust :(
- We can not expect perfect operational security for decades :(
- They can not afford **becoming a target for bad actors** :((

Bad actors capabilities[2] to generate convincing fakes are *expanding fast*

[2] as well as their economic incentives to attempt such attacks

"Just go to The Internet Archive"

There exist **online digital archives** with snapshots of the past 🎉

- They become the new weakest link of the chain of trust :(
- We can not expect perfect operational security for decades :(
- They can not afford **becoming a target for bad actors** :((

Bad actors capabilities[2] to generate convincing fakes are *expanding fast*

We need to fix this now[3] 💣

[2] as well as their economic incentives to attempt such attacks

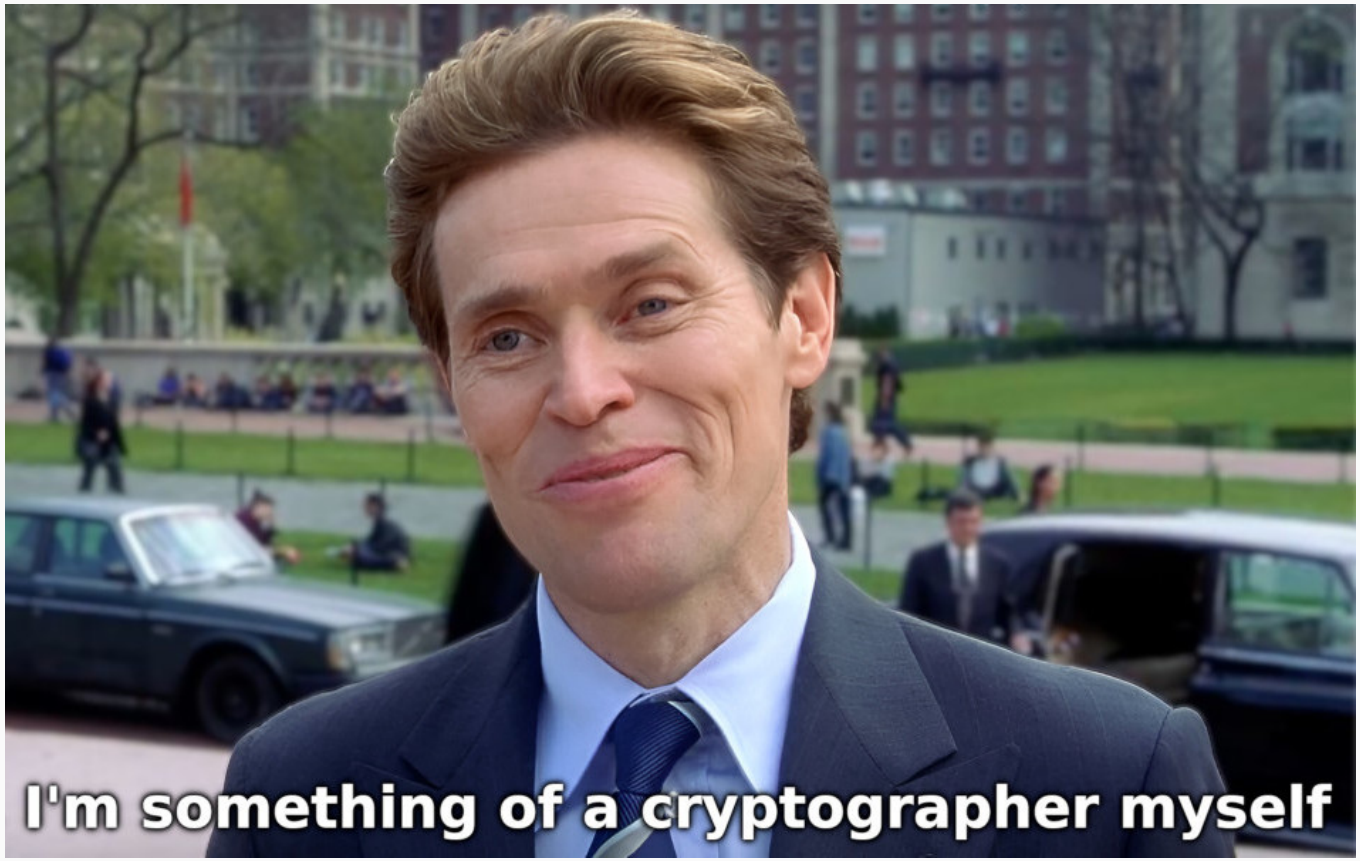
[3] and we should have started 5 years ago

The fix is digital signatures, no?

just add cryptogrophy

The fix is digital signatures, no?

You're in luck!



3. How can we fix this?

What about Trusted Timestamps?

"RFC3161: Internet X.509 PKI Time-Stamp Protocol (TSP)" *2001*

What about Trusted Timestamps?

"RFC3161: Internet X.509 PKI Time-Stamp Protocol (TSP)" *2001*

- A **trusted third-party** signing a "current now" together with a hash[4]

[4] may be used as a *proof of existence* of the hashed data at the signed timestamp

What about Trusted Timestamps?

"RFC3161: Internet X.509 PKI Time-Stamp Protocol (TSP)" *2001*

- A **trusted third-party** signing a "current now" together with a hash[4]
- Used in legal compliance & other *business-for-business* industries

[4] may be used as a *proof of existence* of the hashed data at the signed timestamp

What about Trusted Timestamps?

"RFC3161: Internet X.509 PKI Time-Stamp Protocol (TSP)" *2001*

- A **trusted third-party** signing a "current now" together with a hash[4]
- Other more elaborate constructs are available[5] (blockchains yay!)

[4] may be used as a *proof of existence* of the hashed data at the signed timestamp
[5] we'll first use RFC3161 as "good cryptography" is not our limiting factor here

What about Trusted Timestamps?

"RFC3161: Internet X.509 PKI Time-Stamp Protocol (TSP)" *2001*

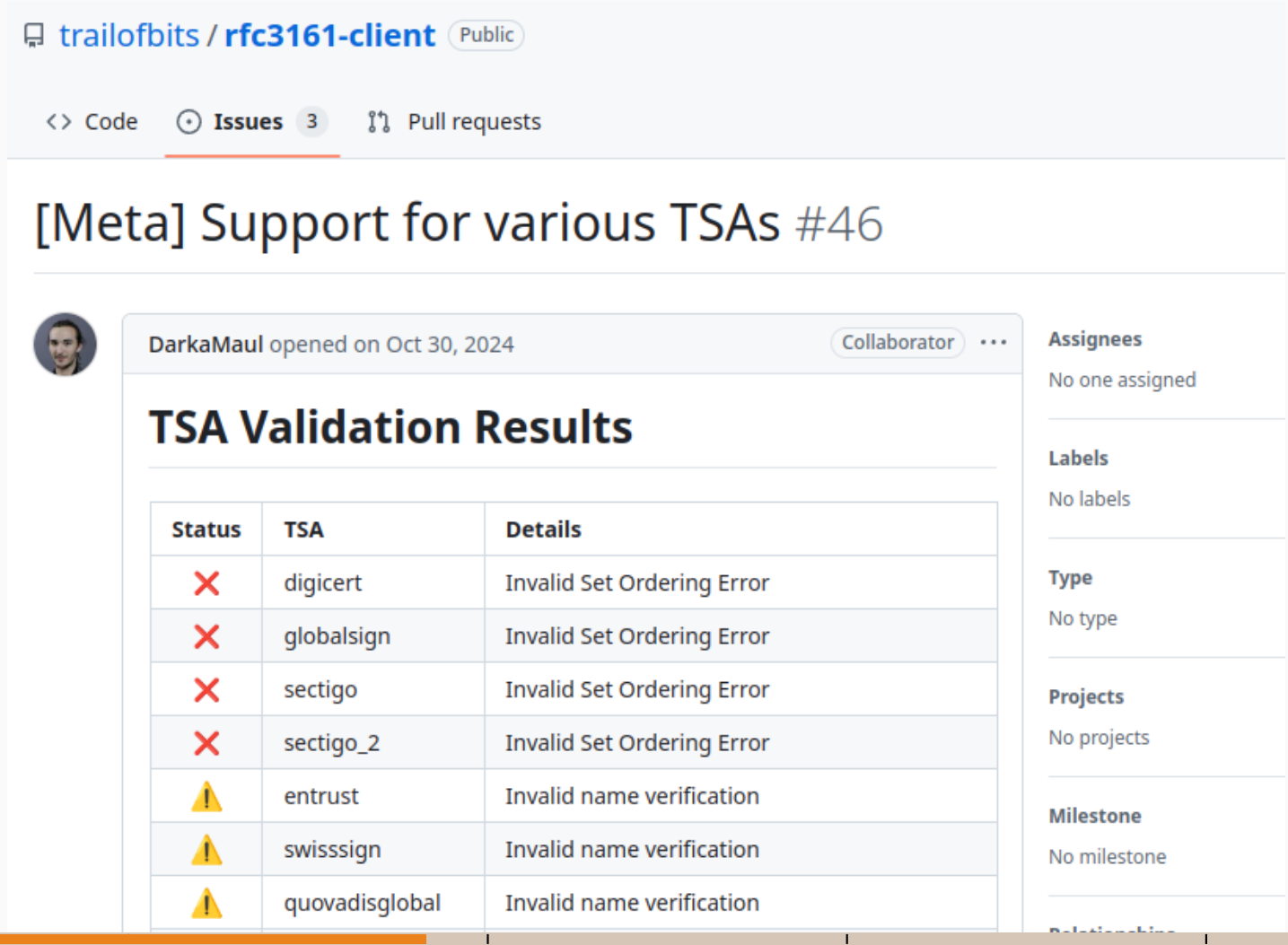
- A **trusted third-party** signing a "current now" together with a hash[4]
- Other more elaborate constructs are available[5] (blockchains yay!)
- **Some public tools** available around trusted timestamps

[4] may be used as a *proof of existence* of the hashed data at the signed timestamp
[5] we'll first use RFC3161 as "good cryptography" is not our limiting factor here

What about Trusted Timestamps?

"RFC3161: Internet X.509 PKI Time-Stamp Protocol (TSP)" *2001*

- A **trusted third-party** signing a "current now" together with a hash[4]
- Other more elaborate constructs are available[5] (blockchains yay!)
- **Some public tools** available around trusted timestamps



What about Trusted Timestamps?

"RFC3161: Internet X.509 PKI Time-Stamp Protocol (TSP)" *2001*

- A **trusted third-party** signing a "current now" together with a hash[4]
- Other more elaborate constructs are available[5] (blockchains yay!)
- **Some public tools** available around trusted timestamps
- Some existing usage in `WebRecorder`[6] – a **web archiving** tool!

[4] may be used as a *proof of existence* of the hashed data at the signed timestamp
[5] we'll first use RFC3161 as "good cryptography" is not our limiting factor here
[6] notably used by `perma.cc` from the Harward Library Innovation Lab

What about Trusted Timestamps?

"RFC3161: Internet X.509 PKI Time-Stamp Protocol (TSP)" *2001*

- A **trusted third-party** signing a "current now" together with a hash[4]
- Other more elaborate constructs are available[5] (blockchains yay!)
- **Some public tools** available around trusted timestamps
- Some existing usage in `WebRecorder`[6] – a **web archiving** tool!

Let's take a look at what they are doing! 🧐

[4] may be used as a *proof of existence* of the hashed data at the signed timestamp
[5] we'll first use RFC3161 as "good cryptography" is not our limiting factor here
[6] notably used by `perma.cc` from the Harward Library Innovation Lab

The missing ”chronology oracle”

(a look at an existing usage of trusted timestamp in digital archives)

1. **Build a web capture** (a WACZ archive) for a document collection

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. **Build a web capture** (a WACZ archive) for a document collection
2. Use your domain certificate[7] to sign an inventory of the documents

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. **Build a web capture** (a WACZ archive) for a document collection
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. **Build a web capture** (a WACZ archive) for a document collection
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

It's simple & sound,

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. **Build a web capture** (a WACZ archive) for a document collection
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

It's simple & sound, **but not really what we want!** 😇

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. ~~**Build a web capture** (a WACZ archive) for a document collection~~
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

It's simple & sound, **but not really what we want!** 😇

We do **NOT**[8] want to store any document ourselves!

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example
[8] trusted timestamps are easier to scale than archival (b/c storage, copyright issues…)

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. ~~**Build a web capture** (a WACZ archive) for a document collection~~
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

It's simple & sound, **but not really what we want!** 😇

We do **NOT**[8] want to store any document ourselves!

- Most documents existing today do not have any signature attached

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example
[8] trusted timestamps are easier to scale than archival (b/c storage, copyright issues…)

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. ~~**Build a web capture** (a WACZ archive) for a document collection~~
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

It's simple & sound, **but not really what we want!** 😇

We do **NOT**[8] want to store any document ourselves!

- Most documents existing today do not have any signature attached
- Is there a **public database of trusted timestamps** of *all the things*?

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example
[8] trusted timestamps are easier to scale than archival (b/c storage, copyright issues…)

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. ~~**Build a web capture** (a WACZ archive) for a document collection~~
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

It's simple & sound, **but not really what we want!** 😇

We do **NOT**[8] want to store any document ourselves!

- Most documents existing today do not have any signature attached
- Is there a **public database of trusted timestamps** of *all the things*?
- Is there an "oracle" somewhere

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example
[8] trusted timestamps are easier to scale than archival (b/c storage, copyright issues…)

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. ~~**Build a web capture** (a WACZ archive) for a document collection~~
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

It's simple & sound, **but not really what we want!** 😇

We do **NOT**[8] want to store any document ourselves!

- Most documents existing today do not have any signature attached
- Is there a **public database of trusted timestamps** of *all the things*?
- Is there an "oracle" somewhere that can answer the "when" & "where"

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example
[8] trusted timestamps are easier to scale than archival (b/c storage, copyright issues…)

The missing "chronology oracle"

(a look at an existing usage of trusted timestamp in digital archives)

1. ~~**Build a web capture** (a WACZ archive) for a document collection~~
2. Use your domain certificate[7] to sign an inventory of the documents
3. Reach out to a public RFC3161 server and get a trusted timestamp

It's simple & sound, **but not really what we want!** 😇

We do **NOT**[8] want to store any document ourselves!

- Most documents existing today do not have any signature attached
- Is there a **public database of trusted timestamps** of *all the things*?
- Is there an "oracle" somewhere that can answer the "when" & "where" based on a huge collection of historical trusted timestamps?

[7] `WebRecorder` own tool uses an ACME-issued Let's Encrypt certificate, for example
[8] trusted timestamps are easier to scale than archival (b/c storage, copyright issues…)

A way to build a ”chronology oracle”

(the way X.509 certificates work today seems to be a good starting point)

A way to build a ”chronology oracle”

(the way X.509 certificates work today seems to be a good starting point)

- a model centered around **independent authorities** signing *stuff*

A way to build a ”chronology oracle”

(the way X.509 certificates work today seems to be a good starting point)

- a model centered around **independent authorities** signing *stuff*
- a third-party acting as observer, signing the ”when” & ”where” online

A way to build a ”chronology oracle”

(the way X.509 certificates work today seems to be a good starting point)

- a model centered around **independent authorities** signing *stuff*
- a third-party acting as observer, signing the ”when” & ”where” online
- these observers would produce as much timestamps as possible *(only)*

A way to build a ”chronology oracle”

(the way X.509 certificates work today seems to be a good starting point)

- a model centered around **independent authorities** signing *stuff*
- a third-party acting as observer, signing the ”when” & ”where” online
- these observers would produce as much timestamps as possible *(only)*
- their output is **manifest** files fit to be stored, distributed & archived

A way to build a "chronology oracle"

(the way X.509 certificates work today seems to be a good starting point)

- a model centered around **independent authorities** signing *stuff*
- a third-party acting as observer, signing the "when" & "where" online
- these observers would produce as much timestamps as possible *(only)*
- their output is **manifest** files fit to be stored, distributed & archived

All the pieces are already there! ⚡

Where are the transparency logs?

transparency in title but not in talk??

4. What about transparency logs?

Transparency logs, you say?

That's a whole new ecosystem you're talking about! 😱

Transparency logs, you say?

That's a whole new ecosystem you're talking about! 😱

- **How are you going to establish trust?**

Transparency logs, you say?

That's a whole new ecosystem you're talking about! 😱

- **How are you going to establish trust?** ...

Transparency logs, you say?

That's a whole new ecosystem you're talking about! 😱

- **How are you going to establish trust?** ... Trust you say?

Transparency logs, you say?

That's a whole new ecosystem you're talking about! 😱

- **How are you going to establish trust?** ... Trust you say?
- *like trust into independent authorities signing certificate-like artifacts?*

Transparency logs, you say?

That's a whole new ecosystem you're talking about! 😱

- **How are you going to establish trust?** ... Trust you say?
- *like trust into independent authorities signing certificate-like artifacts?*

...is this a transparency log we need?

Digital Archive Transparency

Why using **transparency logs in digital archives**?

Digital Archive Transparency

Why using **transparency logs in digital archives**?

- # of new certificates today far exceed # of new documents published

Digital Archive Transparency

Why using **transparency logs in digital archives**?

- # of new certificates today far exceed # of new documents published
- help build a more **"formal" trust model** for independent observers

Digital Archive Transparency

Why using **transparency logs in digital archives**?

- # of new certificates today far exceed # of new documents published
- help build a more **"formal" trust model** for independent observers
  - focus on "when"+"where" as it can be **verified by a third-party**

Digital Archive Transparency

Why using **transparency logs in digital archives**?

- # of new certificates today far exceed # of new documents published
- help build a more **"formal" trust model** for independent observers
  - focus on "when"+"where" as it can be **verified by a third-party**
  - help identify bad observers (sign-now-release-later, other attacks...)

Digital Archive Transparency

Why using **transparency logs in digital archives**?

- # of new certificates today far exceed # of new documents published
- help build a more **"formal" trust model** for independent observers
  - focus on "when"+"where" as it can be **verified by a third-party**
  - help identify bad observers (sign-now-release-later, other attacks...)
- put all manifests[9] in a transparency log to make them **discoverable**

[9] or even all documents ever produced, several billions of them 🚀

Digital Archive Transparency

Why using **transparency logs in digital archives**?

- # of new certificates today far exceed # of new documents published
- help build a more **"formal" trust model** for independent observers
  - focus on "when"+"where" as it can be **verified by a third-party**
  - help identify bad observers (sign-now-release-later, other attacks...)
- put all manifests[9] in a transparency log to make them **discoverable**
- an **existing technology** acting de facto as a "chronology oracle"

[9] or even all documents ever produced, several billions of them 🚀

Digital Archive Transparency

Why using **transparency logs in digital archives**?

- # of new certificates today far exceed # of new documents published
- help build a more **"formal" trust model** for independent observers
  - focus on "when"+"where" as it can be **verified by a third-party**
  - help identify bad observers (sign-now-release-later, other attacks...)
- put all manifests[9] in a transparency log to make them **discoverable**
- an **existing technology** acting de facto as a "chronology oracle"

Let's add a transparency log in our trust model! 🎉

[9] or even all documents ever produced, several billions of them 🚀

Let's build it now!

gonna build it meself

Let's build it now!

gonna build it meself [10]

[10] a very humbling experience in software engineering

5. Where I'm at?
(a work in progress)

Step 1/4: getting the documents

Where to get documents?

- write our own way of downloading the internet

Step 1/4: getting the documents

Where to get documents?

- ~~write our own way of downloading the internet~~
- integrate with existing archiving formats (`warc`/`wacz`, ...)

Step 1/4: getting the documents

Where to get documents?

- ~~write our own way of downloading the internet~~
- integrate with existing archiving formats (`warc`/`wacz`, …)
- get documents from the already-existing[11] digital archives!

[11] most only exposes `md5` or `sha1` in their metadata (unfit for digital signatures…)

Step 1/4: getting the documents

Where to get documents?

- ~~write our own way of downloading the internet~~
- integrate with existing archiving formats (`warc`/`wacz`, ...)
- get documents from the already-existing[11] digital archives!
- start with the documents you have on the local filesystem :)

[11] most only exposes `md5` or `sha1` in their metadata (unfit for digital signatures...)

Step 1/4: getting the documents

done wip started later

Where to get documents?

- ~~write our own way of downloading the internet~~
- integrate with existing archiving formats (`warc`/`wacz`, ...)
- get documents from the already-existing[11] digital archives!
- start with the documents you have on the local filesystem :)

[11] most only exposes `md5` or `sha1` in their metadata (unfit for digital signatures...)

Step 1/4: getting the documents

■ done ■ wip ■ started ■ later

Where to get documents?

- ~~write our own way of downloading the internet~~
- ■ integrate with existing archiving formats (`warc`/`wacz`, …)
- ■ get documents from the already-existing[11] digital archives!
- ■ start with the documents you have on the local filesystem :)

⚠ also collect document metadata for cross-reference ⚠

[11] most only exposes `md5` or `sha1` in their metadata (unfit for digital signatures…)

Step 1/4: getting the documents

done wip started later

Where to get documents?

- ~~write our own way of downloading the internet~~
- integrate with existing archiving formats (`warc`/`wacz`, ...)
- get documents from the already-existing[11] digital archives!
- start with the documents you have on the local filesystem :)

⚠ also collect document metadata for cross-reference ⚠

In most cases, write custom code[12] on a case-by-case basis...

[11] most only exposes `md5` or `sha1` in their metadata (unfit for digital signatures...)
[12] such as some batch PDF processing with some basic metadata extraction

Step 2/4: specifying ”what we sign”

We are not going to sign each document hash+metadata separately...

Step 2/4: specifying ”what we sign”

We are not going to sign each document hash+metadata separately...

- We aggregate data about a document collection in a *manifest file*

Step 2/4: specifying ”what we sign”

We are not going to sign each document hash+metadata separately...

- We aggregate data about a document collection in a *manifest file*
  - pick a format fit for long-term storage & archival (!)

Step 2/4: specifying "what we sign"

We are not going to sign each document hash+metadata separately...

- We aggregate data about a document collection in a *manifest file*
  - pick a format fit for long-term storage & archival (!)
  - **opiniated choice:** ASCII-encoded text files?

Step 2/4: specifying "what we sign"

We are not going to sign each document hash+metadata separately...

- We aggregate data about a document collection in a *manifest file*
  - pick a format fit for long-term storage & archival (!)
  - **opiniated choice:** ASCII-encoded text files?
  - make it human-readable, printable, and fully specified! 💪

[13] see bitcoin's BIP-0173 and BIP-0350

Step 2/4: specifying "what we sign"

We are not going to sign each document hash+metadata separately...

- We aggregate data about a document collection in a *manifest file*
  - pick a format fit for long-term storage & archival (!)
  - **opiniated choice:** ASCII-encoded text files?
  - make it human-readable, printable, and fully specified! 💪
- Lots of hashes & short byte strings to store as text...
  - `base32` with error-correction: `bech32m`-inspired[13] "`r2value`" format

[13] see bitcoin's BIP-0173 and BIP-0350

Step 2/4: specifying "what we sign"

We are not going to sign each document hash+metadata separately...

- We aggregate data about a document collection in a *manifest file*
  - pick a format fit for long-term storage & archival (!)
  - **opiniated choice:** ASCII-encoded text files?
  - make it human-readable, printable, and fully specified! 💪
- Lots of hashes & short byte strings to store as text...
  - `base32` with error-correction: `bech32m`-inspired[13] "`r2value`" format
- we also need structure for resilience to *minor file corruptions*

[13] see bitcoin's BIP-0173 and BIP-0350

Step 2/4: specifying "what we sign"

We are not going to sign each document hash+metadata separately...

- We aggregate data about a document collection in a *manifest file*
  - pick a format fit for long-term storage & archival (!)
  - **opiniated choice:** ASCII-encoded text files?
  - make it human-readable, printable, and fully specified! 💪
- Lots of hashes & short byte strings to store as text...
  - `base32` with error-correction: `bech32m`-inspired[13] "`r2value`" format
- we also need structure for resilience to *minor file corruptions*
- ...

[13] see bitcoin's BIP-0173 and BIP-0350

Step 2/4: specifying "what we sign"

done wip started later

We are not going to sign each document hash+metadata separately...

- We aggregate data about a document collection in a *manifest file*
  - pick a format fit for long-term storage & archival (!)
  - **opiniated choice:** ASCII-encoded text files?
  - make it human-readable, printable, and fully specified! 💪
- Lots of hashes & short byte strings to store as text...
  - `base32` with error-correction: `bech32m`-inspired[13] "`r2value`" format
- we also need structure for resilience to *minor file corruptions*
- ...

What does your "manifest" look like?

[13] see bitcoin's BIP-0173 and BIP-0350

What does these ”manifest” looks like?

a manifest authenticating miscellaneous documents

File /tmp/manifest-r2mh1kejsmkl3n7w37rykcjskm7ssgf4vr5wrvw8yhkulzescssuzcdhq1jmcard.txt

```
  +xapdid:9c99865e-e5d1-483b-924c-98181467f144
  @file:12_April_2023.pdf
  @date:20230502T140309Z
  @title:"CVE Board Meeting Summary - 12 April 2023"
  @author:"CVE&#174; Program"
rip-add pdf 00023 r2ah166fsr0ug92e830rqqqlzffqjkel43gp6cqnj3c6eegqjwszdad0s1h094nw 2334622
  +md5:r2h51d7gped8nuel7qeqn0u46vs577q16uwcvy
  +sha1:r2h01jpypqvyy83s8ddghjlchadl9hphtglyq1efdkvg
  +doi:10.1194/jlr.D085217
  +xapdid:6c0610ac-1dd2-11b2-0a00-9209271dd700
  @file:1301.full.pdf
  @date:20180621T055501Z
  @modify:20200129T133837Z
  @title:"High-throughput, nonperturbing quantification of lipid droplets with digital holographic %
      % microscopy"
rip-add pdf 00024 r2ah1vvzdzsqezmsym03muet6kv2shn2ksg30y70hcz4dhdy9rxptyxqq14j0a7c 1360233
  +md5:r2h51upd2ezxun6s0l4yakae0k8c28514x7nju
  +sha1:r2h017jg997n3kv7dz6arspefh28t6hatp8vp1uwk56m
  @file:13TCASII_replica.pdf
  @date:20140115T110003Z
  @title:"Replica Technique for Adaptive Refresh Timing of Gain Cell embedded DRAM"
rip-add pdf 00025 r2ah1e4s6ctsgafpam35zf4qxm2hn7k7pxu9m3sdpvvfm985mxn8kxzus1fxkulm 472315
  +md5:r2h51ws22dphq54hd2fd9w03c7yg4jy1hkqkmy
  +sha1:r2h01ncumglytgctkvpxerexmppnrea83vg5uh15kn790
```

What does these ”manifest” looks like?

documents are referred to using their sha256 hashes

File /tmp/manifest-r2mh1kejsmkl3n7w37rykcjskm7ssgf4vr5wrvw8yhkulzescssuzcdhq1jmcard.txt

```
  +xapdid:9c99865e-e3d1-483b-924c-98181467f144
  @file:12_April_2023.pdf
  @date:20230502T140309Z
  @title:"CVE Board Meeting Summary - 12 April 2023"
  @author:"CVE&#174; Program"
rip-add pdf 00023 r2ah166fsr0ug92e830rqqqlzffqjkel43gp6cqnj3c6eegqjwszdad0s1h094nw 2334622
  +md5:r2h51d7gped8nuel7qeqn0u46vs577q16uwcvy
  +sha1:r2h01jpypqvyy83s8ddghjlchadl9hphtglyq1efdkvg
  +doi:10.1194/jlr.D085217
  +xapdid:6c0610ac-1dd2-11b2-0a00-9209271dd700
  @file:1301.full.pdf
  @date:20180621T055501Z
  @modify:20200129T133837Z
  @title:"High-throughput, nonperturbing quantification of lipid droplets with digital holographic %
      % microscopy"
rip-add pdf 00024 r2ah1vvzdzsqezmsym03muet6kv2shn2ksg30y70hcz4dhdy9rxptyxqq14j0a7c 1360233
  +md5:r2h51upd2ezxun6s0l4yakae0k8c28514x7nju
  +sha1:r2h017jg997n3kv7dz6arspefh28t6hatp8vp1uwk56m
  @file:13TCASII_replica.pdf
  @date:20140115T110003Z
  @title:"Replica Technique for Adaptive Refresh Timing of Gain Cell embedded DRAM"
rip-add pdf 00025 r2ah1e4s6ctsgafpam35zf4qxm2hn7k7pxu9m3sdpvvfm985mxn8kxzus1fxkulm 472315
  +md5:r2h51ws22dphq54hd2fd9w03c7yg4jy1hkqkmy
  +sha1:r2h01ncumglytqctkvpxerexmpnrea83vq5uh15kn790
```

documents fingerprint (sha256)

What does these ”manifest” looks like?

other identifiers / ”codes” are included for cross-reference

File /tmp/manifest-r2mh1kejsmkl3n7w37rykcjskm7ssgf4vr5wrvw8yhkulzescssuzcdhq1jmcard.txt

```
  +xapdid:9c99865e-e3d1-483b-924c-98181467f144
  @file:12_April_2023.pdf
  @date:20230502T140309Z
  @title:"CVE Board Meeting Summary - 12 April 2023"
  @author:"CVE&#174; Program"
rip-add pdf 00023 r2ah166fsr0ug92e830rqqqlzffqjkel43gp6cqnj3c6eegqjwszdad0s1h094nw 2334622
  +md5:r2h51d7gped8nuel7qeqn0u46vs577q16uwcvy
  +sha1:r2h01jpypqvyy83s8ddghjlchadl9hphtglyq1efdkvg
  +doi:10.1194/jlr.D085217
  +xapdid:6c0610ac-1dd2-11b2-0a00-9209271dd700
  @file:1301.full.pdf
  @date:20180621T055501Z
  @modify:20200129T133837Z
  @title:"High-throughput, nonperturbing quantification of lipid droplets with digital holographic %
      % microscopy"
rip-add pdf 00024 r2ah1vvzdzsqezmsym03muet6kv2shn2ksg30y70hcz4dhdy9rxptyxqq14j0a7c 1360233
  +md5:r2h51upd2ezxun6s0l4yakae0k8c28514x7nju
  +sha1:r2h017jg997n3kv7dz6arspefh28t6hatp8vp1uwk56m
  @file:13TCASII_replica.pdf
  @date:20140115T110003Z
  @title:"Replica Technique for Adaptive Refresh Timing of Gain Cell embedded DRAM"
rip-add pdf 00025 r2ah1e4s6ctsgafpam35zf4qxm2hn7k7pxu9m3sdpvvfm985mxn8kxzus1fxkulm 472315
  +md5:r2h51ws22dphq54hd2fd9w03c7yg4jy1hkqkmy
  +sha1:r2h01ncumglytqctkvpxerexmpprea83vq5uh15kn790
```

other identifiers (or "codes")

What does these ”manifest” looks like?

document metadata is explicitly stored for third-party analysis

File /tmp/manifest-r2mh1kejsmkl3n7w37rykcjskm7ssgf4vr5wrvw8yhkulzescssuzcdhq1jmcard.txt

```
  +xapdid:9c99865e-e3d1-483b-924c-98181467f144
  @file:12_April_2023.pdf
  @date:20230502T140309Z
  @title:"CVE Board Meeting Summary - 12 April 2023"
  @author:"CVE&#174; Program"
rip-add pdf 00023 r2ah166fsr0ug92e830rqqqlzffqjkel43gp6cqnj3c6eegqjwszdad0s1h094nw 2334622
  +md5:r2h51d7gped8nuel7qeqn0u46vs577q16uwcvy
  +sha1:r2h01jpypqvyy83s8ddghjlchadl9hphtglyq1efdkvg
  +doi:10.1194/jlr.D085217
  +xapdid:6c0610ac-1dd2-11b2-0a00-9209271dd700
  @file:1301.full.pdf
  @date:20180621T055501Z
  @modify:20200129T133837Z
  @title:"High-throughput, nonperturbing quantification of lipid droplets with digital holographic %
      % microscopy"
rip-add pdf 00024 r2ah1vvzdzsqezmsym03muet6kv2shn2ksg30y70hcz4dhdy9rxptyxqq14j0a7c 1360233
  +md5:r2h51upd2ezxun6s0l4yakae0k8c28514x7nju
  +sha1:r2h017jg997n3kv7dz6arspefh28t6hatp8vp1uwk56m
  @file:13TCASII_replica.pdf
  @date:20140115T110003Z
  @title:"Replica Technique for Adaptive Refresh Timing of Gain Cell embedded DRAM"
rip-add pdf 00025 r2ah1e4s6ctsgafpam35zf4qxm2hn7k7pxu9m3sdpvvfm985mxn8kxzus1fxkulm 472315
  +md5:r2h51ws22dphq54hd2fd9w03c7yg4jy1hkqkmy
  +sha1:r2h01ncumglytqctkvpxerexmpnrea83vq5uh15kn790
```

other document metadata (title, date, ...)

Step 3/4: getting a trusted timestamp & signing the manifest

We have everything in a manifest, **we still need some signatures!**

Step 3/4: getting a trusted timestamp & signing the manifest

We have everything in a manifest, **we still need some signatures!**

- first we get a `rfc3161` trusted timestamp of the manifest content

Step 3/4: getting a trusted timestamp & signing the manifest

We have everything in a manifest, **we still need some signatures!**

- first we get a `rfc3161` trusted timestamp of the manifest content
  - use `rfc3161-client` & `rfc3161ng` to process query & responses

Step 3/4: getting a trusted timestamp & signing the manifest

We have everything in a manifest, **we still need some signatures!**

- first we get a `rfc3161` trusted timestamp of the manifest content
  - use `rfc3161-client` & `rfc3161ng` to process query & responses
  - validate against 34 (!) public TSA[14] and `openssl-ts` as reference

[14] Time-Stamp Authority – each with a subtly unique & non-standard behavior ✨

Step 3/4: getting a trusted timestamp & signing the manifest

We have everything in a manifest, **we still need some signatures!**

- first we get a `rfc3161` trusted timestamp of the manifest content
  - use `rfc3161-client` & `rfc3161ng` to process query & responses
  - validate against 34 (!) public TSA[14] and `openssl-ts` as reference
- then we append a signature[15] bound to a given observer identity

[14] Time-Stamp Authority – each with a subtly unique & non-standard behavior ✨

[15] likely some hybrid `ed25519`+`MAYO3` digital signature scheme (*wip pqc yadiyada*)

Step 3/4: getting a trusted timestamp & signing the manifest

We have everything in a manifest, **we still need some signatures!**

- first we get a `rfc3161` trusted timestamp of the manifest content
  - use `rfc3161-client` & `rfc3161ng` to process query & responses
  - validate against 34 (!) public TSA[14] and `openssl-ts` as reference
- then we append a signature[15] bound to a given observer identity
- then add some tooling (like a CLI) to *work around all that*

[14] Time-Stamp Authority – each with a subtly unique & non-standard behavior ✨

[15] likely some hybrid `ed25519`+`MAYO3` digital signature scheme (*wip pqc yadiyada*)

Step 3/4: getting a trusted timestamp & signing the manifest

done wip started later

We have everything in a manifest, **we still need some signatures!**

- first we get a `rfc3161` trusted timestamp of the manifest content
  - use `rfc3161-client` & `rfc3161ng` to process query & responses
  - validate against 34 (!) public TSA[14] and `openssl-ts` as reference
- then we append a signature[15] bound to a given observer identity
- then add some tooling (like a CLI) to *work around all that*

[14] Time-Stamp Authority – each with a subtly unique & non-standard behavior ✨

[15] likely some hybrid `ed25519`+`MAYO3` digital signature scheme (*wip pqc yadiyada*)

Step 3/4: getting a trusted timestamp & signing the manifest

done wip started later

We have everything in a manifest, **we still need some signatures!**

- first we get a `rfc3161` trusted timestamp of the manifest content
  - use `rfc3161-client` & `rfc3161ng` to process query & responses
  - validate against 34 (!) public TSA[14] and `openssl-ts` as reference
- then we append a signature[15] bound to a given observer identity
- then add some tooling (like a CLI) to *work around all that*

We now have a digitally-signed "certificate-like way"

[14] Time-Stamp Authority – each with a subtly unique & non-standard behavior ✨

[15] likely some hybrid `ed25519`+`MAYO3` digital signature scheme (*wip pqc yadiyada*)

Step 3/4: getting a trusted timestamp & signing the manifest

done wip started later

We have everything in a manifest, **we still need some signatures!**

- first we get a `rfc3161` trusted timestamp of the manifest content
  - use `rfc3161-client` & `rfc3161ng` to process query & responses
  - validate against 34 (!) public TSA[14] and `openssl-ts` as reference
- then we append a signature[15] bound to a given observer identity
- then add some tooling (like a CLI) to *work around all that*

We now have a digitally-signed "certificate-like way"
to archive & share the "when" & "where"

[14] Time-Stamp Authority – each with a subtly unique & non-standard behavior ✨
[15] likely some hybrid `ed25519`+`MAYO3` digital signature scheme (*wip pqc yadiyada*)

Step 4/4: distributing the result & establishing trust

Time to supercharge this with transparency logs! 🚀

Step 4/4: distributing the result & establishing trust

Time to supercharge this with transparency logs! 💥

sorry, this is future works[16]

[16] insert `meme_not_much_but_honest_work.jpg`

Conclusion & Thank you

To sum it up:

- we need timestamping services **signing everything at scale** 📈 (fast)

Conclusion & Thank you

To sum it up:

- we need timestamping services **signing everything at scale** 📈 (fast)
- **cheap to operate** with good tooling to be accessible *to anyone*

Conclusion & Thank you

To sum it up:

- we need timestamping services **signing everything at scale** 📈 (fast)
- **cheap to operate** with good tooling to be accessible *to anyone*
- the technology already exists and is **ready to be used**

Conclusion & Thank you

To sum it up:

- we need timestamping services **signing everything at scale** 📈 (fast)
- **cheap to operate** with good tooling to be accessible *to anyone*
- the technology already exists and is **ready to be used**
- *we "just" need to build the thing* – and I'm trying to! [17]

[17] but I'm short on staff & resources – come see me after that talk! 💪

Conclusion & Thank you

To sum it up:

- we need timestamping services **signing everything at scale** 📈 (fast)
- **cheap to operate** with good tooling to be accessible *to anyone*
- the technology already exists and is **ready to be used**
- *we "just" need to build the thing* – and I'm trying to! [17]

Transparency logs are a perfect fit for our use case!

[17] but I'm short on staff & resources – come see me after that talk! 💪

Conclusion & Thank you

To sum it up:

- we need timestamping services **signing everything at scale** 📈 (fast)
- **cheap to operate** with good tooling to be accessible *to anyone*
- the technology already exists and is **ready to be used**
- *we "just" need to build the thing* – and I'm trying to! [17]

Transparency logs are a perfect fit for our use case!

To keep you posted, write to:
contact@archive.rip

thank you for listening! 

[17] but I'm short on staff & resources – come see me after that talk! 💪

Appendix: Links & Resources

Several resources for the curious mind:

- The Century Scale Storage as an introduction to digital archiving
- The LOCKSS / CLOCKSS project as precursor in distributed archives
- The WACZ Signing & Verification from WebRecorder for reference
- Both Signed HTTP Exchanges & HTTP Message Signatures 🎶
- and many more I'm sure :)