

HUMBOLDT-UNIVERSITÄT ZU BERLIN  
MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT  
INSTITUT FÜR INFORMATIK

# **Towards a Generalized Framework for Secure Time-Stamping**

Masterarbeit

zur Erlangung des akademischen Grades  
Master of Science (M. Sc.)

eingereicht von: Keno Goertz  
geboren am: 20.11.1999  
geboren in: Saarlouis

Gutachter/innen: Prof. Dr. Matthias Weidlich  
Prof. Dr. Florian Tschorsch

eingereicht am: ..... verteidigt am: .....

**Abstract.** Write your abstract here.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Trusted time-stamping . . . . .	1
1.2	Distributed trust . . . . .	1
1.2.1	Building trust through publication . . . . .	1
1.2.2	Quantifying distributed trust . . . . .	2
1.2.3	Increasing availability . . . . .	6
1.2.4	Protecting against Byzantine failures . . . . .	9
1.3	Implementations of time-stamp publication . . . . .	11
1.3.1	Witness signatures . . . . .	11
1.3.2	Random-witness time-stamping . . . . .	12
1.3.3	Threshold cryptography . . . . .	12
	<b>References</b>	<b>13</b>
	<b>Glossary</b>	<b>13</b>
	<b>Acronyms</b>	<b>13</b>
	<b>Appendix</b>	<b>i</b>

# 1 Introduction

## 1.1 Trusted time-stamping

The simplest approach to digital time-stamping relies on a trusted third party (TTP). If Alice wants to time-stamp a document and prove the document's existence at the time-stamp's time to Bob at some later time, she can ask a time-stamp authority (TSA) to cryptographically sign a secure hash of her document together with the current time. Bob accepts the TSA's signature as proof of the document's existence at the specified time.<sup>1</sup>

From now on, I will refer to the Alice, the party requesting a time-stamp from the TSA, as the *document owner*. Bob, the party who wants to verify a time-stamp's validity, will be called the *verifier*.

The scheme outlined above requires complete trust of both the document owner and the verifier in the impartiality of the TSA. The verifier needs to trust the TSA to keep its private key secure and to never produce time-stamps for the past (an attack which I will refer to as *backdating*). The document owner needs to trust the availability of the time-stamping service provided by the TSA whenever she wants to time-stamp a document.

This trust in a single authority can be problematic in practice. Even if we could assume complete impartiality of the TSA with regard to the document owner and the verifier, what happens if the party responsible for running the TSA wants to time-stamp a document of their own? Clearly, to ensure impartiality, another TSA would need to be used. But now what if neither of our TSAs can be assumed to be impartial with regard to yet another party who wants to time-stamp a document? Manually keeping track of which TSA can be trusted under which circumstances quickly becomes impractical. The notion of distributed trust will simplify matters considerably.

## 1.2 Distributed trust

### 1.2.1 Building trust through publication

Trusted time-stamping requires complete trust in the time-stamp authority. This does not mean, however, that the TSA is actually *trustworthy*. We can decrease the amount of trust that we need to put in any single party by distributing trust across multiple parties.

In the context of time-stamping, we can achieve this by requiring the TSA to *publish* its time-stamps to a large number of *witnesses*. The publication can be implemented in many different ways, which we will take a detailed look at in Section 1.3. For now, the reader may imagine that the TSA publishes its time-stamps in a newspaper.

---

<sup>1</sup>Stuart Haber and W. Scott Stornetta (Jan. 1991). "How to time-stamp a digital document". In: *Journal of Cryptology* 3.2, pp. 99–111. ISSN: 1432-1378. DOI: 10.1007/BF00196791.

The time-stamping company *Surety* actually employed this method of publication in practice.<sup>2</sup>

Witnesses keep a record of the time-stamps issued by the TSA. They do not accept time-stamps issued too far in the past. Staying with the example of time-stamps published in a newspaper, the newspaper archives of public libraries can act as witnesses. To prevent backdating attacks, a library only archives a newspaper which it receives on the printed date of publication.

When a verifier wants to confirm the validity of a time-stamp, she can now ask a selection of witnesses for confirmation. Using our example of newspaper archives, a verifier visits a handful of library archives and confirms that the time-stamp in question is actually printed in the archived newspapers of that date. Verifiers only accept time-stamps for which they find a sufficient number of witnesses.

Using such a publication scheme, a malicious TSA can no longer carry out a backdating attack all by itself. Instead, it would require the active cooperation of a sufficiently large number of witnesses in order to convince a verifier of the validity of a backdated time-stamp. The verifier's trust is thus *distributed* over the TSA, the publication process and the witnesses.

### 1.2.2 Quantifying distributed trust

Let us now introduce a mathematical model for the publication scheme outlined in Section 1.2.1. Say the TSA publishes its time-stamps to  $N$  witnesses. It should be emphasized that a witness is required to keep a record of time-stamps. Going back to our example of time-stamps published in a newspaper,  $N$  does *not* correspond to the number of copies printed. Instead,  $N$  refers to the number of places that keep archives of the newspaper.

We assume that there exist a number  $K$  of malicious witnesses that collude together with the TSA in an attempt to backdate time-stamps.

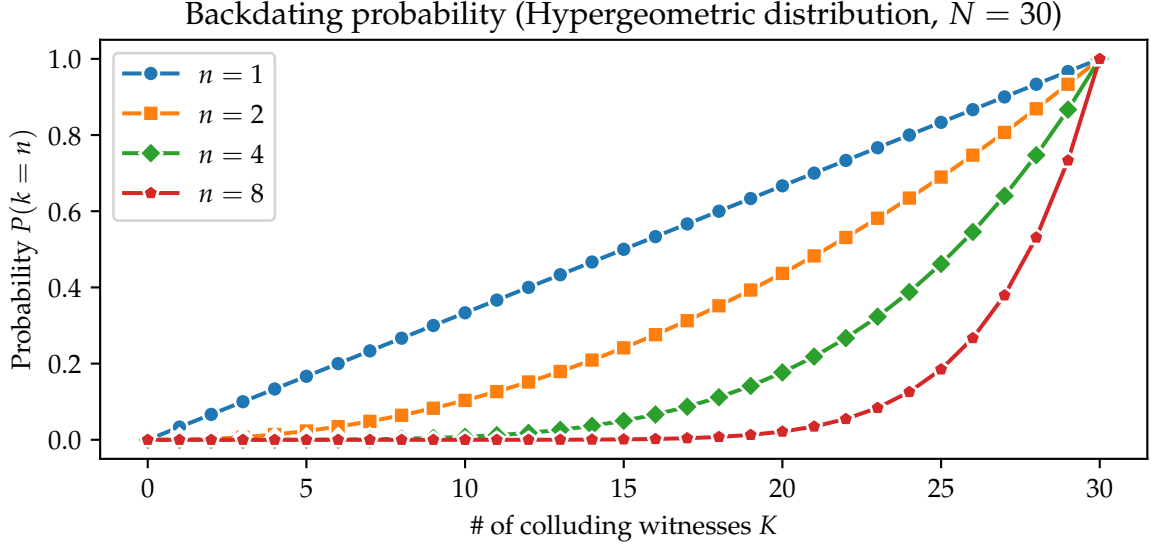
Finally, a verifier consults a number  $n$  of witnesses to verify a time-stamp. The verifier only accepts the time-stamp if all  $n$  selected witnesses confirm its existence at the given time.

Let  $k$  be the number of maliciously colluding witnesses selected by the verifier. Evidently, a successful backdating attack occurs when the verifier selects only colluding witnesses, so when  $k = n$ .

Let us now further assume that the verifier selects its  $n$  witnesses from the total number of witnesses  $N$  at random with a uniform distribution. Our problem is now equivalent to the urn problem when "drawing without replacement".  $k$  thus follows

---

<sup>2</sup>"As an extra measure, Surety publishes a weekly summary hash value in The New York Times. This 'widely-witnessed' value provides an anchor for the security of the whole system." Surety LLC (n.d.). *What We Do*. <https://web.archive.org/web/20250325081455/https://www.surety.com/digital-copyright-protection/prove-ownership>. Accessed: 25 March 2025



**Figure 1:** Probability of a successful backdating attack according to the hypergeometric distribution.  $N = 30$  witnesses keep records of the time-stamps issued by the TSA. Of these witnesses, a number  $K$  (plotted on the x-axis) maliciously collude with the TSA in order to backdate time-stamps. To check a time-stamp's validity, a verifier consults  $n$  randomly selected witnesses. The backdating attack is successful if all  $n$  selected witnesses are malicious. As expected, the probability of a successful backdating attack increases with an increasing number of colluding witnesses  $K$ , reaching 1 when  $N = K$ . The verifier can decrease the likelihood of a successful backdating attack by consulting more witnesses, as can be observed from the different graph lines.

the hypergeometric distribution<sup>3</sup> with the probability mass function:

$$\text{hypg}(k; n, K, N) = \binom{K}{k} \binom{N-K}{n-k} / \binom{N}{n} \quad (1)$$

The probability of a successful backdating attack is then given by the equation:

$$P(k = n) = \text{hypg}(n; n, K, N) = \binom{K}{n} / \binom{N}{n} \quad (2)$$

Figure 1 graphs this probability as a function of  $K$  for different values of  $n$ .

In practice, the selection of witnesses may not be truly random. Sticking to our example of newspaper archives, a verifier might prefer libraries which are geographically close to them. A network protocol for distributed trust may also favor witnesses with small round-trip times in order to increase performance.

An attacker may be able to leverage this by placing colluding witnesses at favorable locations. We can model this by introducing a weight parameter  $\omega$ , where a verifier

<sup>3</sup>Catherine Forbes et al. (Nov. 2010). *Statistical Distributions*. 4th ed. Wiley-Blackwell, pp. 117-119.

is  $\omega$  times more likely to select a malicious witness than an honest witness.  $k$  then follows a noncentral hypergeometric distribution.

Two distinct noncentral hypergeometric distributions exist in the literature. They are frequently confused, as the difference between them is subtle and both are regularly referred to as “the” noncentral hypergeometric distribution.<sup>4</sup> Fisher’s noncentral hypergeometric distribution models the case where multiple balls are drawn from the urn at once and thus the probability of drawing one item is independent of the other items that are drawn. The sample size  $n$  can not be known in advance in this case, as this would introduce a dependence between draws. Wallenius’ noncentral hypergeometric distribution, on the other hand, models the case of sequentially drawing balls from the urn, for a total number of  $n$  draws that has been determined in advance.<sup>5</sup>

As the verifier in our model determines the number  $n$  of witnesses to consult in advance,  $k$  follows Wallenius’ noncentral hypergeometric distribution. The verifier selects witnesses in rounds.  $k_\nu$  describes how many malicious witnesses she selects after the completion of round  $\nu$ . The probability of selecting a malicious witness in round  $\nu + 1$  corresponds to the weight ratio of the remaining witnesses:

$$p_{\nu+1} = \frac{(K - k_\nu)\omega}{(K - k_\nu)\omega + (N - K) - (n - k_\nu)} \quad (3)$$

The probability mass function for  $k$  after selecting all  $n$  witnesses is:

$$\text{wnchypg}(k; n, K, N, \omega) = \binom{K}{k} \binom{N - K}{n - k} \cdot \int_0^1 \left(1 - t^{\omega/d}\right)^k \left(1 - t^{1/d}\right)^{n-k} dt \quad (4)$$

$$d = (K - k)\omega + (N - K) - (n - k) \quad (5)$$

The probability of a successful backdating attack is then:

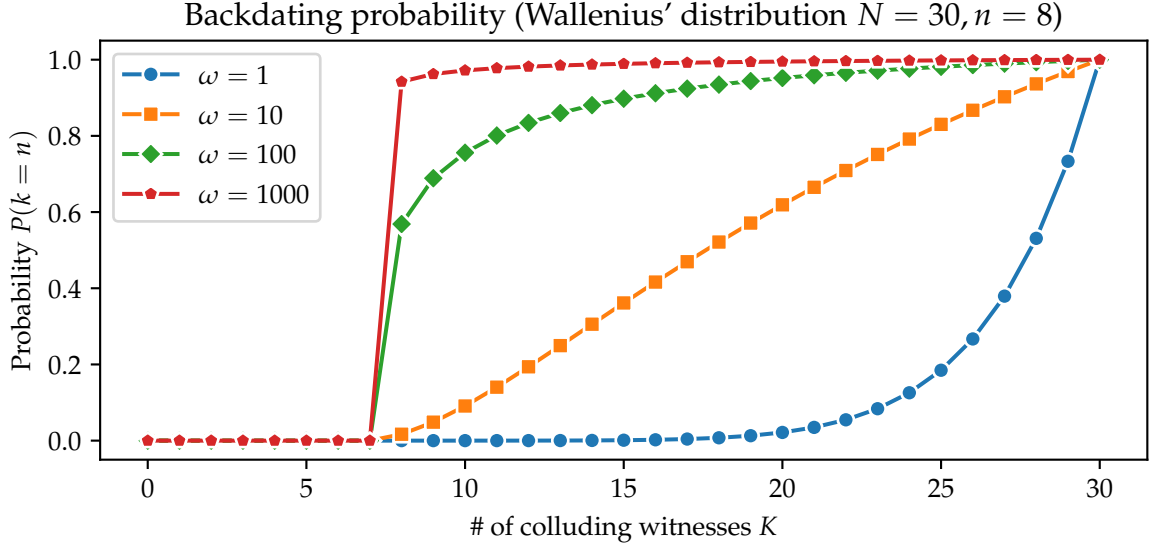
$$P(k = n) = \text{wnchypg}(n; n, K, N, \omega) = \binom{K}{n} \cdot \int_0^1 \left(1 - t^{\omega/((K-n)\omega + N-n)}\right)^n dt \quad (6)$$

Figure 2 graphs this probability as a function of  $K$  for different values of  $\omega$ .

Note that the noncentral hypergeometric distribution is equivalent to the regular hypergeometric distribution when  $\omega = 1$ . When an attacker can ensure that the verifier

<sup>4</sup>Agner Fog (2008). “Calculation Methods for Wallenius’ Noncentral Hypergeometric Distribution”. In: *Communications in Statistics - Simulation and Computation* 37.2, pp. 258–273. DOI: 10.1080/03610910701790269.

<sup>5</sup>For a detailed discussion on the distinction between Wallenius’ and Fisher’s noncentral hypergeometric distribution, see: Agner Fog (2008). “Calculation Methods for Wallenius’ Noncentral Hypergeometric Distribution”. In: *Communications in Statistics - Simulation and Computation* 37.2, pp. 258–273. DOI: 10.1080/03610910701790269



**Figure 2:** Probability of a successful backdating attack according to Wallenius' noncentral hypergeometric distribution.  $N = 30$  witnesses keep records of the time-stamps issued by the TSA. Of these witnesses, a number  $K$  (plotted on the x-axis) maliciously collude with the TSA in order to backdate time-stamps. To check a time-stamp's validity, a verifier consults  $n = 8$  randomly selected witnesses. The verifier is  $\omega$  times more likely to select a malicious witness than an honest witness. The backdating attack is successful if all  $n$  selected witnesses are malicious. As expected, the probability of a successful backdating attack increases with an increasing number of colluding witnesses  $K$ , reaching 1 when  $N = K$ . Increasing values of  $\omega$  increase the chances of a successful backdating attack, as can be observed from the different graph lines. For  $\omega = 1$ , the graph matches the hypergeometric distribution of Fig. 1. For large values of  $\omega$ , the graph approaches a step function with the step at  $K = n = 8$ .



will only select malicious witnesses,  $\omega$  approaches infinity. In this case, the probability of a successful backdating attack approaches a step function with the step at  $n = K$ .

$$\lim_{\omega \rightarrow \infty} \text{wnchypg}(n; n, K, N, \omega) = \begin{cases} 0 & n < K \\ 1 & n \geq K \end{cases} \quad (7)$$

### 1.2.3 Increasing availability

In a real distributed service, we can not assume that a verifier can always reach any witness she desires. Network partitions or denial of service attacks may render witnesses temporarily unavailable. We include a new parameter  $n'$  into our model to accomodate this possibility. While the verifier still asks  $n$  randomly selected witnesses to verify a time-stamp, she accepts the time-stamp as soon as she receives  $n'$  valid responses from the witnesses, with  $n' < n$ .

Let  $U$  be the total number of witnesses that are unavailable or refuse to confirm a legitimate time-stamp upon a client's request. Let  $u$  be the number of unavailable witnesses included in the  $n$  witnesses that the verifier randomly selected. The verifier will then not accept a legitimate time-stamp if  $u > n - n'$ . The probability of this happening according to the hypergeometric distribution is:

$$P(u > n - n') = \sum_{u=n-n'+1}^n \binom{U}{u} \binom{N-U}{n-u} / \binom{N}{n} \quad (8)$$

Figure 3 graphs this probability as a function of  $U$  for different values of  $n'$ .

If a verifier is more likely to select certain witnesses over others and we assume that an attacker can carry out a targeted denial of service attack on these witnesses, we need to model the probability of a successful DoS attack using Wallenius' noncentral hypergeometric distribution:

$$P(u > n - n') = \sum_{u=n-n'+1}^n \binom{U}{u} \binom{N-U}{n-u} \quad (9)$$

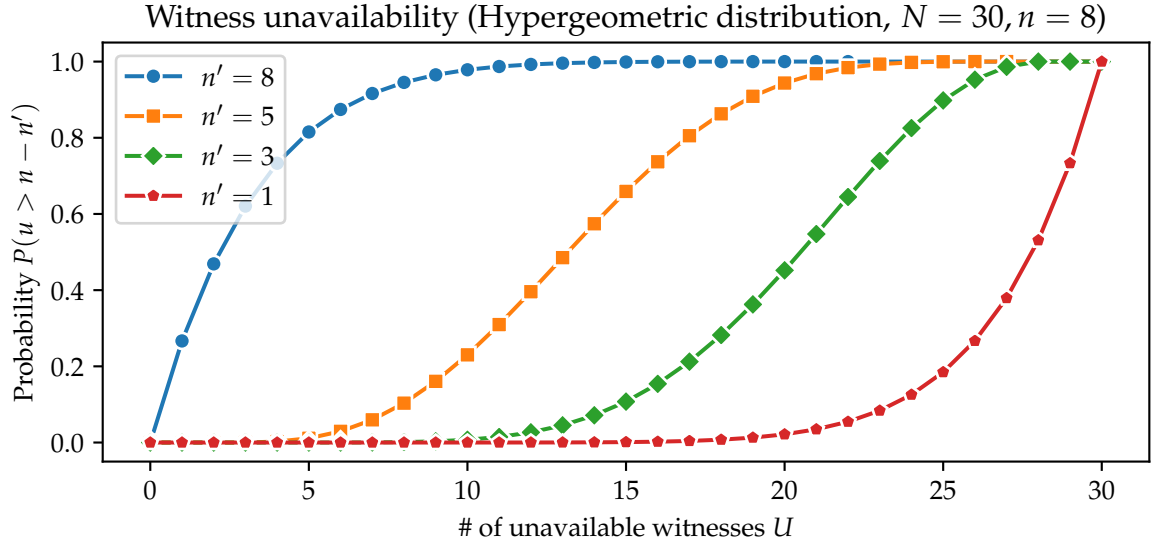
$$\cdot \int_0^1 \left(1 - t^{\omega/d(u)}\right)^u \left(1 - t^{1/d(u)}\right)^{n-u} dt \quad (10)$$

$$d(u) = (U - u)\omega + (N - U) - (n - u)$$

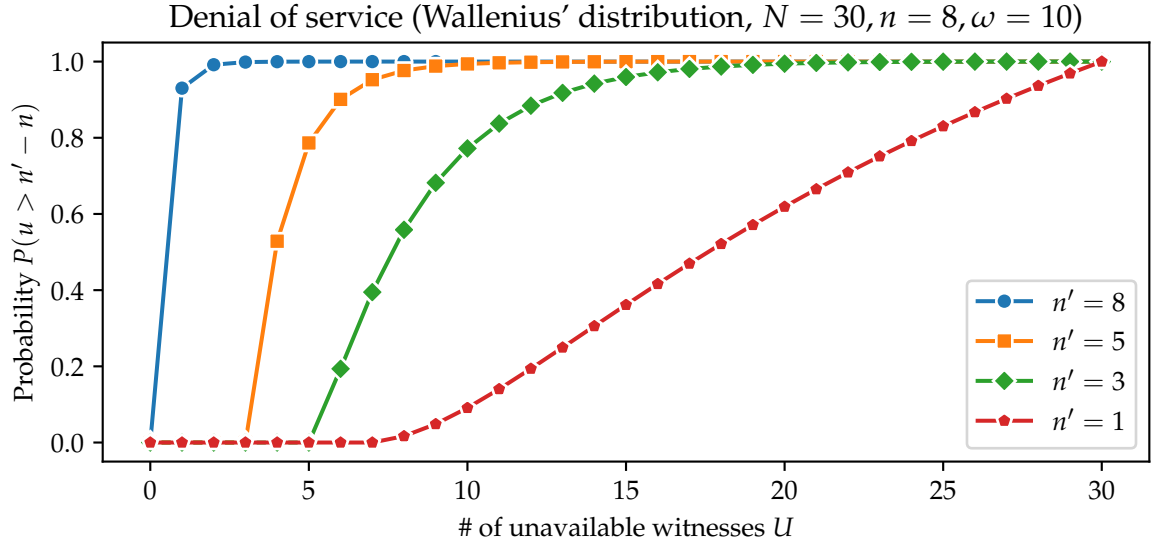
Figure 4 graphs this probability as a function of  $U$  for different values of  $n'$ .

While the introduction of  $n'$  increases availability in the face of network partitions or denial of service attacks, it compromises the security against backdating attacks. A backdating attack is now successful when  $k \geq n'$ .

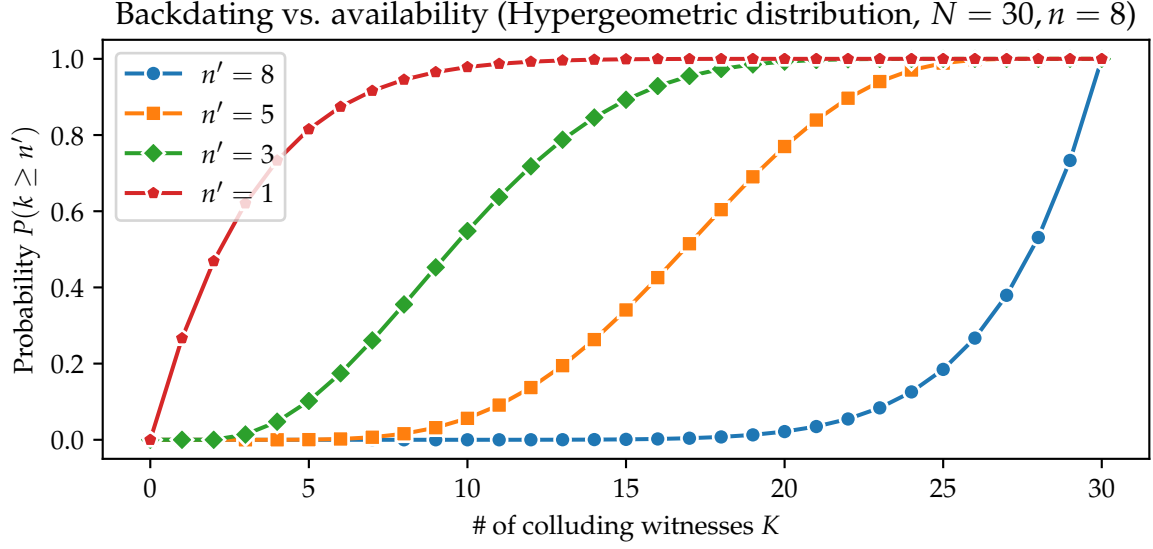
In the case of the hypergeometric distribution, this leaves us with the following equation.



**Figure 3:** Probability of a verifier failing to accept a legitimate time-stamp in the face of witness unavailability.  $N = 30$  witnesses keep records of the time-stamps issued by the TSA. Of these witnesses, a number  $U$  (plotted on the x-axis) is unavailable due to a network partition, a denial of service attack, a crash failure or some other reason. To check a time-stamp's validity, a verifier consults  $n = 8$  randomly selected witnesses. She accepts the time-stamp if she receives valid responses from  $n'$  witnesses. The verifier will fail to accept a legitimate time-stamp if more than  $n - n'$  of the selected witnesses are unavailable. Decreasing values of  $n'$  protect against this happening, as can be observed from the different graph lines.



**Figure 4:** Probability of a verifier failing to accept a legitimate time-stamp in the face of a targeted denial of service attack.  $N = 30$  witnesses keep records of the time-stamps issued by the TSA. Of these witnesses, a number  $U$  (plotted on the x-axis) is unavailable due to a targeted DoS attack. To check a time-stamp's validity, a verifier consults  $n = 8$  randomly selected witnesses. The verifier is  $\omega = 10$  times more likely to select an unavailable witness than an available witness. She accepts the time-stamp if she receives valid responses from  $n'$  witnesses. The verifier will fail to accept a legitimate time-stamp if more than  $n - n'$  of the selected witnesses are unavailable. Decreasing values of  $n'$  protect against DoS attacks, as can be observed from the different graph lines.



**Figure 5:** Probability of a successful backdating attack according to the hypergeometric distribution when allowing witness unavailability.  $N = 30$  witnesses keep records of the time-stamps issued by the TSA. Of these witnesses, a number  $K$  (plotted on the x-axis) maliciously collude with the TSA in order to backdate time-stamps. To check a time-stamp's validity, a verifier consults  $n = 8$  randomly selected witnesses. She accepts the time-stamp if she receives valid responses from  $n'$  witnesses. The backdating attack is successful if at least  $n'$  of the selected witnesses are malicious. Decreasing values of  $n'$  increase the chances of a successful backdating attack, as can be observed from the different graph lines.

$$P(k \geq n') = \sum_{k=n'}^n \binom{K}{k} \binom{N-K}{n-k} / \binom{N}{n} \quad (11)$$

Figure 5 graphs this probability as a function of  $K$  for different values of  $n'$ .

The probability of a successful backdating attack according to Wallenius' distribution is then:

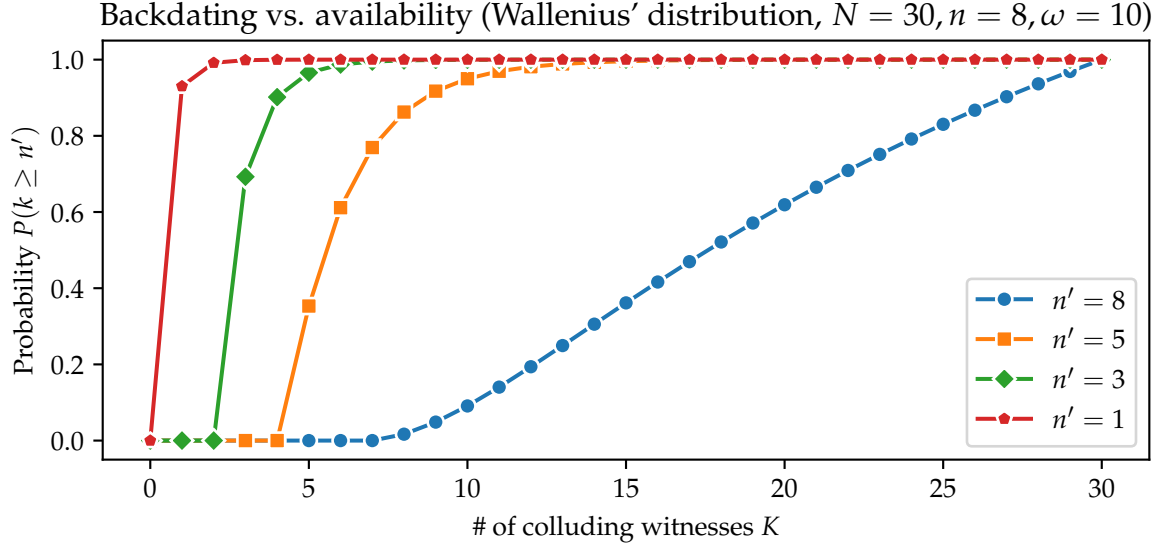
$$P(k \geq n') = \sum_{k=n'}^n \binom{K}{k} \binom{N-K}{n-k} \cdot \int_0^1 \left(1 - t^{\omega/d(k)}\right)^k \left(1 - t^{1/d(k)}\right)^{n-k} dt \quad (12)$$

$$d(k) = (K - k)\omega + (N - K) - (n - k) \quad (13)$$

Figure 6 graphs this probability as a function of  $K$  for different values of  $n'$ .

#### 1.2.4 Protecting against Byzantine failures

We can regard both witness unavailability and the malicious collusion of witnesses for a backdating attack as types of Byzantine failures. Let  $B$  be the number of Byzantine



**Figure 6:** Probability of a successful backdating attack according to Wallenius' noncentral hypergeometric distribution when allowing witness unavailability.  $N = 30$  witnesses keep records of the time-stamps issued by the TSA. Of these witnesses, a number  $K$  (plotted on the x-axis) maliciously collude with the TSA in order to backdate time-stamps. To check a time-stamp's validity, a verifier consults  $n = 8$  randomly selected witnesses. She accepts the time-stamp if she receives valid responses from  $n'$  witnesses. The verifier is  $\omega = 10$  times more likely to select a malicious witness than an honest witness. The backdating attack is successful if at least  $n'$  of the selected witnesses are malicious. Decreasing values of  $n'$  increase the chances of a successful backdating attack, as can be observed from the different graph lines.

witnesses. Full protection against backdating as well as denial of service attacks is provided by the system if and only if:

$$n' > B \quad (\text{Protection against backdating}) \quad (14)$$

$$n \geq n' + B > 2B \quad (\text{Protection against DoS}) \quad (15)$$

If  $n \leq 2B$ , it is impossible to guarantee protection against both failure modes. In this case, there exists a fundamental trade-off concerning the choice of  $n'$ . Higher values provide better protection against backdating attacks, while lower values better protect against DoS.

If the choice of  $n$  does not guarantee protection against Byzantine failures, it is important that the verifier randomly selects witnesses without bias. If the verifier favors certain witnesses ( $\omega > 1$ ), this can vastly increase the chances of a successful attack, as can be observed by comparing Figure 3 with Figure 4, or Figure 5 with Figure 6.

### 1.3 Implementations of time-stamp publication

#### 1.3.1 Witness signatures

In Section 1.2, we have modeled time-stamp publication in a rather traditional way. In our model, the TSA publishes its time-stamps to  $N$  witnesses, of which a verifier later consults a number  $n$  to confirm the validity of a time-stamp. Time-stamps need to be published to all  $N$  participating witnesses, who are required to keep records of all valid time-stamps they encounter. Such a scheme can even be implemented by publishing time-stamps in a newspaper.

We can improve the scheme by having witnesses cryptographically sign the time-stamp to confirm its validity.<sup>6</sup> The signature serves as a verifiable record of the witnessing. Witnesses can send the signed time-stamp to the document owner, which frees them from the responsibility of keeping records. After all, it is sensible that the party requesting a time-stamp should be responsible for storing the time-stamp and the information necessary for verification. This scheme has the additional advantage of reducing the communication cost for time-stamp verification. When a verifier wants to confirm a time-stamp, she only needs to communicate with the document owner, when before she had to exchange messages with  $n$  witnesses.

These modifications don't influence the statistical models described in Section 1.2. All equations of this section are still applicable.

We can also reduce the communication cost of time-stamp publication by only asking  $n$  witnesses for signatures instead of publishing the time-stamp to all  $N$

---

<sup>6</sup>Dave Bayer, Stuart Haber, and W. Scott Stornetta (1993). "Improving the Efficiency and Reliability of Digital Time-Stamping". In: *Sequences II*. ed. by Renato Capocelli, Alfredo De Santis, and Ugo Vaccaro. Springer New York, pp. 329–334. ISBN: 978-1-4613-9323-8.

participating witnesses. This still offers full protection against backdating and denial of service attacks if  $n$  and  $n'$  are chosen according to Equations (14) and (15).

If  $n \leq 2B$ , some mechanism should be used to ensure that witnesses are chosen randomly with uniform distribution ( $\omega = 1$ ). As explained in Section 1.2.4, this minimizes the probability of a successful backdating or denial of service attack. Section 1.3.2 describes one such mechanism.

Finally, it should be noted that witness signatures are really just time-stamps. This allows us to get rid of the TSA altogether. A document owner can directly send her document hash and the current time to  $n$  witnesses and collect the returning signatures. By sending  $n'$  witness signatures to a verifier, the document owner can prove the validity of her time-stamp.

### 1.3.2 Random-witness time-stamping

Haber and Stornetta proposed using a pseudo-random number generator (PRNG) to ensure uniformly distributed random witness selection for the purpose of distributed time-stamping.<sup>7</sup> Each participating witness is initially assigned a unique identifier. The document owner can then seed the PRNG with the hash of her document and interpret the PRNG's output as witness identifiers. This way, she selects the  $n$  witnesses responsible for signing her time-stamp. To confirm the time-stamp's validity, a verifier first checks the witness signatures. She then verifies that the  $n$  identifiers produced by the PRNG when seeded with the document's hash are a superset of the identifiers corresponding to the  $n'$  witness signatures.

This scheme is secure if the hash function applied to the document has the *one-way* property: Given a desired output hash, it should be computationally hard to find an input for which the hash function produces this output. If the hash function did not have this property, a document owner could possibly carry out a backdating attack by colluding with at least  $n'$  witnesses. She would be able to construct a meaningful document such that the witnesses selected by the PRNG would be those colluding with her ( $\omega \rightarrow \infty$ ).

Another security requirement is that the PRNG produces uniformly distributed identifiers. A non-uniform distribution could again potentially be exploited ( $\omega > 1$ ) to increase the probability of a successful backdating or DoS attack.

The random-witness scheme proposed by Haber and Stornetta is desirable if we are not sure that Equations (14) and (15) hold, and hence want to ensure  $\omega = 1$  to minimize the probability of successful backdating and DoS attacks.

### 1.3.3 Threshold cryptography

---

<sup>7</sup>Stuart Haber and W. Scott Stornetta (Jan. 1991). "How to time-stamp a digital document". In: *Journal of Cryptology* 3.2, pp. 99–111. ISSN: 1432-1378. DOI: 10.1007/BF00196791.

## References

- Bayer, Dave, Stuart Haber, and W. Scott Stornetta (1993). "Improving the Efficiency and Reliability of Digital Time-Stamping". In: *Sequences II*. Ed. by Renato Capocelli, Alfredo De Santis, and Ugo Vaccaro. Springer New York, pp. 329–334. ISBN: 978-1-4613-9323-8.
- Fog, Agner (2008). "Calculation Methods for Wallenius' Noncentral Hypergeometric Distribution". In: *Communications in Statistics - Simulation and Computation* 37.2, pp. 258–273. DOI: 10.1080/03610910701790269.
- Forbes, Catherine et al. (Nov. 2010). *Statistical Distributions*. 4th ed. Wiley-Blackwell.
- Haber, Stuart and W. Scott Stornetta (Jan. 1991). "How to time-stamp a digital document". In: *Journal of Cryptology* 3.2, pp. 99–111. ISSN: 1432-1378. DOI: 10.1007/BF00196791.
- LLC, Surety (n.d.). *What We Do*. <https://web.archive.org/web/20250325081455/https://www.surety.com/digital-copyright-protection/prove-ownership>. Accessed: 25 March 2025.

## Glossary

- Backdating** An attack trying to forge time-stamps dated in the past
- Document Owner** The party owning the document for which they request a time-stamp
- Time-Stamp Authority** A trusted third party in the context of time-stamping
- Verifier** The party who wants to confirm the validity of a time-stamp
- Witness** An entity who witnessed a timestamp at the time of its creation

## Acronyms

- DoS** Denial of Service
- PRNG** Pseudo-Random Number Generator
- TSA** Time-Stamp Authority
- TTP** Trusted Third Party



# Appendix

## Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den 1. Januar 1970

.....