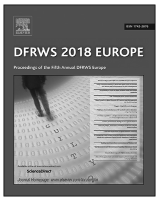
Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin



DFRWS 2018 Europe — Proceedings of the Fifth Annual DFRWS Europe

The reliability of clocks as digital evidence under low voltage conditions

Jens-Petter Sandvik [a, b, *], André Årnes [a, c]

[a] Norwegian University of Science and Technology (NTNU), Norway
[b] National Criminal Investigation Service (Kripos), Norway
[c] Telenor ASA, Norway

Keywords:
Digital forensics
Mobile forensics
Clock
Clock documentation
Low voltage errors

ABSTRACT

Battery powered electronic devices like mobile phones are abundant in the world today, and such devices are often subject to digital forensic examinations. In this paper, we show that the assumptions that clocks are close to correct can be misleading under some circumstances, especially with failing batteries. One of four tested devices showed the clock jumped 8 and 12 years into the future when the battery connector voltage was held at 2.030 V and 2.100 V for about 9 s. Other devices showed a more expected behavior, where the clocks were slowly lagging until it was reset. In addition to this, we tested the precision of some methods of documenting the clock settings, and found most timestamps to be within reasonable precision for forensic use. Finally, we describe a model for the variability of the timestamps examined.

© 2018 The Author(s). Published by Elsevier Ltd on behalf of DFRWS. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Introduction

A forensic investigation often includes questions and hypotheses about the temporal domain, such as questions about when, and in which order certain events happened. It is an established practice in digital forensics to document the clock settings of digital equipment that will be used for evidence, as the trustworthiness of timestamps in the evidence is often questioned in court.

An ideal clock is an oscillator that increases a counter, in the same frequency as other clocks. A physical clock is not ideal, and will be affected by its physical properties, its environment, adjustments, etc. Wander, as the ntp version 4 specification calls it, is the average error over time a clock experiences which degrades the accuracy over time, and jitter is the small perturbations that limits the precision of the clock (Martin et al., 2010).

During normal operations with no external interference to the device, this is how the device works. During its lifetime, a device might be exposed to various conditions that can affect the correctness of operations. One such condition might be a failing battery.

A battery will power the phone until it reaches the lower limit for what the device needs for normal operations. At this point, the device will shut down, but the battery will continue to power the clock. In the end, the battery reaches a level where the energy required to power the low-power clock is insufficient, and the device will shut completely down. The intuitive understanding of the clock's behavior is that the clock will continue to stay close to correct until the memory doesn't have enough electric power to keep its data; then, the processor shuts down and the clock is reset to the device's epoch.

The background for these experiments comes from a case where the logs from a mobile phone indicated that the clock might have been adjusted 24 h forward right before or during the time it was powered off. Observation of the device three years later showed that removing the battery and reinserting it made the clock jump forward by up to 1 min compared to the original clock. The test was done using a power supply attached to the phone's battery pins instead of the battery, as the original battery had degraded to the point that it couldn't power the phone any more. The possibility that the clock had been automatically set into the future could therefore not be ruled out.[1]

* Corresponding author. National Criminal Investigation Service (Kripos), Norway.
E-mail addresses: jens.p.sandvik@ntnu.no, jens-petter.sandvik@politiet.no (J.-P. Sandvik), andrearn@pvv.ntnu.no (A. Årnes).

[1] The case is from the Norwegian court case LB-2016-112427. A timestamp analysis were presented in court and challenged, and a more thorough examination found the origin of the clock adjustments to be inconclusive.

https://doi.org/10.1016/j.diin.2018.01.003
1742-2876/© 2018 The Author(s). Published by Elsevier Ltd on behalf of DFRWS. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

A clock that can be adjusted back or forward in time by naturally occurring error situations, is not a common hypothesis that is considered during investigations, and the failure of considering this can be that an automatic adjustment of a clock is believed to be a manual, willed adjustment of the clock. A search for events at a particular time might return incorrect results because of erroneous timestamps.

While the focus in this paper is on mobile phones, the issues discussed here are also applicable to iot. The total number of future IoT devices vary greatly between estimates, but it is widely believed that the number will be greater than today. As the number of devices increases, the cost goes down, and more devices will operate without supervision. From that, we can conclude that the number of malfunctioning devices, such as devices with failing batteries, outside the supported operating temperature or in other unexpected states will also increase, leading to more errors.

In this paper we propose a hypothesis that the clock of a device can jump back or forward in time when the device experiences a low voltage state on the battery connectors. The alternative hypothesis is that the clock will either be close to correct, or be reset to the device's epoch, depending on the voltage applied. To test this, we first established the precision of various methods and timestamps that can be used for comparing clocks, then we continued to test 4 different mobile phones during low voltage states.

The rest of the paper is structured as follows: First related work is discussed, then a model for delays between the clock and measured timestamp is defined. The experimental setup is then described, followed by the results. Lastly, we round up with a discussion, conclusion, and suggest future work.

Related work

It is generally accepted that the data that makes up evidence can be uncertain and contain errors introduced either by system faults, or humans. A system might malfunction, thereby recording erroneous data, and humans can interfere and change data both willfully and by accident. Abstraction layers can hide the precision of the information, and the interpretation of the data can be uncertain. Casey (2002) described a system for assessing evidence based on its uncertainty by attaching certainty levels to each piece of evidence. Together with the uncertainty of origin, correctness of logs, loss of information, and errors, temporal uncertainty is one of the types of uncertainty that was described during interpretation and reconstruction. This experiment goes a step toward quantifying uncertainties in digital investigations. The fact that digital information might contain errors and uncertainties has highlighted a need for quantifying these possible errors and uncertainties (Erbacher, 2010).

Other research has focused on timestamps and how to detect temporal irregularities. The implications of not considering the timezone that the timestamp references can be serious when presented in court. Boyd and Forster (2004) showed by a case example in which misinterpretations influenced the hypotheses in a case. Kaart and Laraghy (2014) discussed how investigators can detect erroneously set time zones and which time zone the among various timestamps adheres to phones. They also described the configurations related to the clock and timestamps in Android phones.

The use of timestamps are important in investigations for many reasons: to search or carve for information within a certain period, to establish the order of events, or to find usage patterns (Årnes et al., 2017). Willassen (2008a) shows that by using a hypothesis-based approach, investigators can establish a hypothesis about how particular clocks have been adjusted, and test whether the timestamps support or refute the clock hypothesis. The clock hypothesis covering adjustments of a clock can be tested by looking at causally linked events and timestamps outside the possible set of timestamps for a particular clock. The formalism of the proposed method has been used for detecting clock adjustments using the Master File Table in the NTFS file system (Willassen, 2008b).

The memory cells in a processor can be affected by the supply voltages, the temperature, or other external sources like radiation. As the supply voltage to the cells are closing in to the threshold voltage of the transistors, the error rate of the memory cells increases many orders of magnitude (Dreslinski et al., 2010). CMOS circuit performance, among them memory cells, are dependent on many variables, and, e.g., aging processes degrade transistors and increase their threshold voltage (Santos et al., 2016).

To induce faults in processing hardware is not a new technique. Barenghi et al. (2009) exploited the faults happening during a constant low voltage state to attack an RSA software algorithm in an ARM-9 processor using three attacks. When keeping the voltage low on one of the power lines to the System-on-Chip (SoC), the core power line, the LOAD instructions were affected by bit failures. The result of the fault was either data corruption or an instruction swap, both of which could be exploited. Another method of manipulating the Trust Zone in an ARM processor was demonstrated by manipulating the software control of the Dynamic Voltage and Frequency Scaling system in the processor. By manipulating the processor voltage and frequency, cryptographic keys could be extracted from the Trust Zone (Tang et al., 2017).

Timestamp variability

There are many sources of variability in the process leading up to a timestamp or a documentation of the relationship between clocks. In this paper, we propose a model for these variable sources where the granularity of the model can be adjusted as needed. Here, we analyze at different scenarios that all have slightly different delay models. These scenarios are:

- Documenting the timestamp with a photo, using a camera clock for comparison
- Comparison of two clocks from two sources, typically a system clock and a external clock

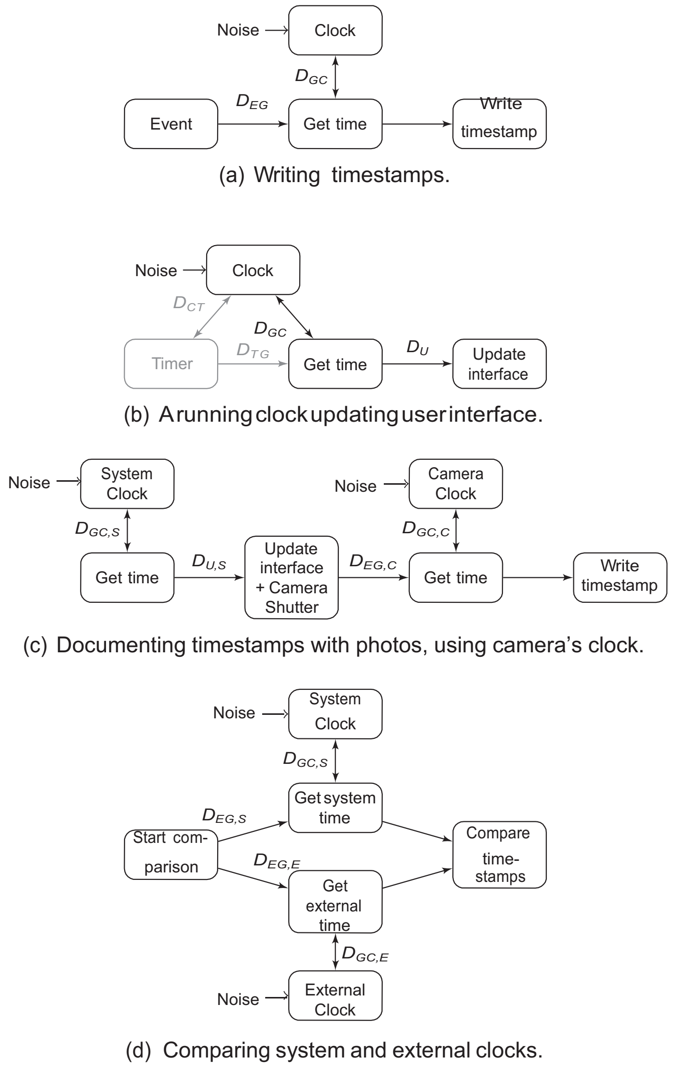
These two scenarios consist of two basic models. These are:

- A timestamp set because of a particular event
- The update of a timestamp in a user interface

We can differentiate between a noise source, which can skew the resulting timestamp in either direction, and a delay that can only increase the difference forward in time between the clock and the resulting timestamp. Fig. 1(a) and (b) shows these two basic models for the source of the inaccuracy of the timestamps. Fig. 1(c) and (d) shows the two scenarios for documenting the time.

Each clock in the model has a noise source attached. In this model, this shows as a summary of all the noise affecting a clock source. These sources tend to be small, but might differ between clocks (Zhou and Nicholls, 2008). A processor can have multiple clock sources, and use the less precise, power saving clock during its sleep state, while it uses a more precise and power intensive clock during normal operations (Qualcomm Technologies, 2016).

In this paper, the formalism presented by Willassen (2008a) is utilized. It states that an event, $e \in E$, where E is the domain of events, can be mapped to the time domain, $t(e), E \rightarrow T$, and furthermore to a timestamp by a clock, c: $c(t(e)) = \tau_c(e), E \rightarrow V$, where V is the domain of time values. The clock function can then



(a) Writing timestamps.

(b) A running clock updating user interface.

(c) Documenting timestamps with photos, using camera's clock.

(d) Comparing system and external clocks.

Fig. 1. A simplified model of timestamp delays and variability. First two figures shows the basic model, the last two combine these into scenarios.

be seen as the sum of an ideal clock and the deviation from that ideal clock:

$$c(t) = b(t) + d(t) \tag{1}$$

where $b(t)$ is the ideal clock at time t and $d(t)$ is the deviation from this clock at the given time, typically an adjustment offset from the correct clock. We use t as a short form for $t(e)$ here.

We expand this definition of d to also include the delay between an event and the clock retrieval, such that:

$$c(t) = b(t) + d(t) + \sum_X D_X(t) \tag{2}$$

where $\sum_X D_X(t)$ is the sum of all delays and noise included in mapping an event to a time value at a particular time. The delays and offsets are values that change with time. A clock can, e.g., be adjusted back in time, and then at a later time forward to correct time. This would affect d in the equation, while an interrupt of the process recording events can affect one of the D_x variables.

Looking at each of the four figures in 1, the first two shows the basic models for timestamps and the two last the scenarios mentioned above. Fig. 1(a) shows a timestamp set as a response to an event: There is a delay, D_{EG}, between the event happening and the call to a function creating a timestamp. This function can be, e.g., "currentTimeMillis ()" or "elapsedRealTime ()" in an Android system. Next is the delay within the function itself querying the clock for its current value, D_{GC}. How small these delays are is dependent on many factors: the time it takes to call a function, context switches happening during the call, interrupts, etc. The timestamps for an event is given by equation c: $\tau_c(t) = b(t) + d(t) + \sum_X D_X(t)$, where d is the offset from the ideal clock, d_X is the noise, D_{EG} and D_{GC} in the figure.

The second model, shown in Fig. 1(b) is when a user interface gets updated. On the device, it will run either a hardware or OS timer, or a timer in an application. The timer is dependent on the clock, and a delay might be introduced in the communication between the timer and the clock, here depicted as D_{CT}. There might be a delay between the time a timer starts and the processing of the function for getting the current time, D_{TG}, and the delay between this function and the query of the clock is the same as the previous scenario, D_{GC}. Finally there is a new delay introduced with the updating of the user interface, D_U. This can be a computer screen, 7-segment display, or other type of interface. This delay is a combination of both the program execution and the physical properties of the display, like refresh rates, setup times for electric signals, etc. The timer circuit is depicted in grey in the figure to show that the delay introduced by the timer process is not affecting the reading of a timestamp from a display, as only the time from the clock that is read to the interface is updated. When the event happens at the user interface update, the clock will be given by $\tau_c(t) = b(t) + d(t) - \sum_X D_X(t)$, where $d(t)$ is the clock offset, and $\sum_X D_X(t)$ is the sum of the noise, D_{GC} and D_U at the given time.

Comparing and documenting clocks with a photo is a combination of the previous two models. Fig. 1(c) shows how this combination can be viewed. The difference in delays between the clocks can be expressed as:

$$\Delta d(t) = c_s(t) - c_c(t) \tag{3}$$

$$= \left(d_s(t) + \sum D_s(t)\right) - \left(d_c(t) + \sum -D_c(t)\right) \tag{4}$$

$$= d_s(t) - d_c(t) + \sum D_s(t)) + \sum D_c(t) \tag{5}$$

where d are the offsets from the ideal clock, and $\sum D$ is the sum of the noise and delays in the respective system at a given time, except for the clock's offset. The time of the event, t, is the update of the display at the same time as the camera's shutter is opening and closing. The delays according to D_c is negative, as these happen after the event that is recorded.

The last scenario, as depicted in Fig. 1(d), is the comparison of two clocks. This is similar to the first scenario, where an event triggers a call for reading the clock. The difference here is that the same event will query two different clocks, and the delays between these two are usually different. In case there is a network clock that is queried, there is an added delay to $D_{GC,E}$ by the network. The differences between these timestamps can be expressed as:

$$\Delta d(t) = d_S(t) + \sum D_S(t) - \left(d_E(t) + \sum D_E(t)\right) \tag{6}$$

$$= d_S(t) - d_E(t) + \sum D_S(t) - \sum D_E(t) \tag{7}$$

The previous points show how delays can be introduced in the time between an event is happening and the clock is retrieved, or from the clock is retrieved to it is shown in an interface. In addition to the sources discussed, bit errors in the memory storing the current value of the clock can be affected by low voltages. In Willassens model, that is extended here, it is the clock offset from the correct clock, $d(t)$ that is affected by a such error leading to a jump in the offset from a correct clock.

Experimental setup

The goal of this work is to establish how clocks in various devices are affected by a very low voltage on the battery poles.

In order to examine the clocks in the tested devices, we first assess the precision of various methods of measuring the clock offset.

The clock sources used in these experiments are either the internal clock of the device, the clock in the GSM network, the clock signal from GPS, or the clock synchronized with the network of NTP servers.

Precision of measuring clocks

To measure the time, two methods were used: documenting both a known clock and a measured clock with photos, and comparing the clocks by connecting the device to another clock and juxtaposing these programatically. Two cameras were used: one Canon 60D SLR camera and one Huawei P9 phone camera. In addition to these, a Linux computer running NTP was used as a presumably correct clock. Additionally, an ordinary sports stopwatch were used to complement the cameras. The use of cameras for documentation of what is shown on a screen is common, giving a timestamp from the camera's clock together with a photographed clock.

A computer that were running Linux and NTP was set up, and the watch command with the -p option was used to update a timestamp every 0.1 s, which then were the limit for the precision of this clock.

A photo of the NTP clock and the running stopwatch was taken with each of the two cameras approximately every 30 s, during a 5 min period. The timestamps were recorded and compared in each photo. This was repeated 10 times, and for each run, the difference between the various timestamps were calculated and the median centered at 0, as it was the variability of the timestamps that were of interest. The datasets for each variable difference were then merged. A list of the various timestamps and their corresponding resolutions is given in Table 1.

For comparing clocks programatically, the Android app ClockSync was used to compare the system clock with an online clock source. The clocks were compared 100 times on each phone in order to establish the precision of this method.

A hypothesis regarding human behavior will need a lower time resolution than a case where malware automatically perform actions on a network. From experience, a subsecond precision for documenting the clock is seldom needed. Events recorded with the same clock will retain their order, while events recorded with different clocks will need to compare the clocks with the resolution needed for deciding their order.

Table 1
Timestamps resolutions.

Timestamp	Resolution
Stop watch	1/100 s
Timestamp embedded in filename	1 s
Last modified in file system	1 s
Timestamps from EXIF	1/1000 s
GPS timestamp in EXIF	1 s
Computer NTP clock shown in terminal	1/10 s
ClockSync App	1/1000 s

Test setup for low voltage test

In order to detect how apparently deterministic systems perform when assumptions about the operating environment fail, these experiments establish the behavior of the internal clock of a device under low voltage conditions. The voltage in these tests was so low that the normal operation of the device was interrupted, and seemed to be turned off. This is comparable to a system where the battery, main or secondary, is near depleted. The battery on the device was removed, and replaced with a adjustable power supply, with a voltage resolution of 1 mV.

The devices tested were selected to be used devices from various vendors, without any stored personal data.

Each experiment were done by first finding the values for the voltage and time where the clock went from operating normally to being reset. This was done by adjusting the voltage, and for each voltage setting testing various times under low voltage.

The hypothesis was that between the “fully powered” state and “not powered” state, the device's clock could be adjusted forward or backward in time. The alternative hypothesis is therefore that between these two states, the clock will either approach correctness or be adjusted to the platform's epoch. To use the formalism introduced: The alternative hypothesis states that d would only be affected as a negative value, and to a predefined epoch. The values D_X will only be responsible for small delays.

The “fully powered” voltage was set to 3.75–3.8 V for all devices, the same value as the specification of the device's original battery.

Results

The results follow the same structure as the experimental setup. First shown are the results from the timestamps related to comparison of clocks followed by the various phones when exposed to low voltages.

Measurement precision in photos

The first test was to establish the precision of the clocks that can be used for measurement. As described the results of 10 runs with about 10 photos in each run was centered to median 0 and combined. Table 2 shows the spread of the various timestamps compared to each other. Fig. 2 visualizes the data as a boxplot where the whiskers are 1.5 times the interquartile range. Samples outside this are shown as outliers in the plot.

For all pictures, the Exif timestamps “SubSecCreateDate”, “SubSecDateTimeOriginal”, and “SubSecModifyDate” were equal, and is just referenced as EXIF metadata.

The results show that the stopwatch is the least variable with regard to the NTP clock. It had the lowest spread and the least difference between the extreme points in the difference with the NTP clock. One of the challenges discovered when taking photos of the stop watch was the difficulty in interpreting the number shown depending on the exposure time used on the camera. In this case, the 7-segment display didn't show the number properly. Especially the 1/100-digit was very inaccurate because of this, but also the 1/10th digit were sometimes hard to categorize.

The clock of the SLR camera has two sets of timestamps, both from the same clock source: the file system timestamp and the Exif

Table 2
Comparison of timestamps. Results centered to median = 0.

#	n	Min	1st Qu.	3rd Qu.	Max
	Phone GPS − NTP				
1.	103	−0.80	−0.20	0.25	3.90
	Phone Filename − NTP				
2.	118	−0.85	−0.20	0.20	0.80
	Phone GPS − Filename				
3.	103	−1	0	0	4
	Phone GPS − EXIF metadata				
4.	103	−0.877	−0.208	0.235	4.074
	Phone EXIF metadata − NTP				
5.	118	−0.25	−0.058	0.053	0.982
	Phone FS modified − NTP				
6.	118	−0.85	−0.20	0.20	1.30
7.	Camera EXIF metadata − NTP				
	118	−0.770	−0.169	0.170	0.605
8.	Camera FS modified − NTP				
	118	−1.30	−0.30	0.29	1.80
	NTP − Stopwatch				
9.	268	−0.08	−0.04	0.02	0.12

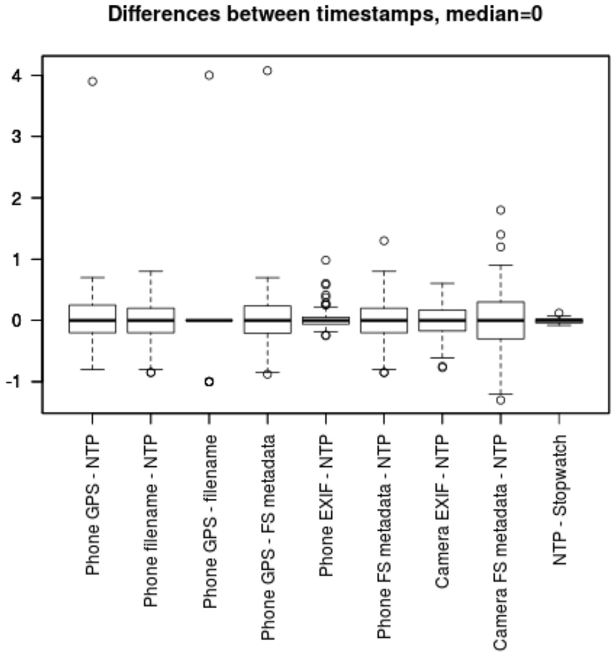


Fig. 2. Comparison of timestamps. Results centered to median = 0.

timestamps. The file system timestamp has a 1 s resolution and is not as precise as the EXIF timestamp.

The phone had three clock sources: the phone's quartz crystal, the network time, and the GPS. The results show that the various timestamps on the phone were variable with regard to each other. The most precise timestamp measured against the NTP clock was the Exif timestamp, which had a spread between the 3rd and 1st quartile of 0.111 s. The timestamps from the GPS information on the pictures had up to 4 s difference with other clocks, something that indicates that this timestamp is not as accurate as the GPS signal itself.

Measurement precision from time server

The app ClockSync from Google Play store was tested to see how precise it could measure the system clock. This is an app that queries the system clock and an online time server, and shows the difference between these timestamps. The precision of the app was tested by recording 100 readings within a few minutes. Table 3 shows the results of running this app on various devices, and Fig. 3 shows a box plot of the same data. As the variability of the measurement method is of interest, the data has been centered to median = 0.

Some of the differences in the precision of the results is a result of different resolutions among the timestamps. The resolution of the timestamps are listed in Table 1.

Clocks on ASUS memopad 10 tablet

The ASUS Memopad 10 tablet is running Android 4.2.2, and the CPU is a ARMv7 Quad core chip from Rockwell (RK101/RK30xx). The device, like many mobile devices, does not have it's own secondary battery, but rather powers the clock during its “off” state from the main battery. The normal battery voltage is 3.8 V, which was used during the high voltage state.

The time was measured with the app ClockSync, and the time discrepancy between the system clock and online time server were measured before and after the period of low voltage. Fig. 4 shows the difference between system clock and NTP clock when varying the time during low voltage state for voltages 2.030 V and 2.100 V. The plot shows that the clock barely changes with regard to the NTP clock, until it has been under low voltage for 9–10 s. In the flat area, the clock is slowing linearly, an indication that the clock stops running, retaining its state, and continues when it has enough power. For 2.030 V, the clock was lagging 2.77 s after 3.63 s, and lagging 8.07 s after 9.02 s. At between 9 and 11 s, the clock is first

Table 3
Precision of ClockSync app that compares system time with NTP time. Results are centered to median = 0.

Device	n	Min.	1st Qu.	3rd Qu.	Max
P9	100	−0.048	−0.002	0.003	0.030
Memopad	102	−0.266	−0.086	0.157	0.503
Y625	100	−0.216	−0.027	0.004	0.249
Xcover 2	100	−0.041	−0.003	0.002	0.034

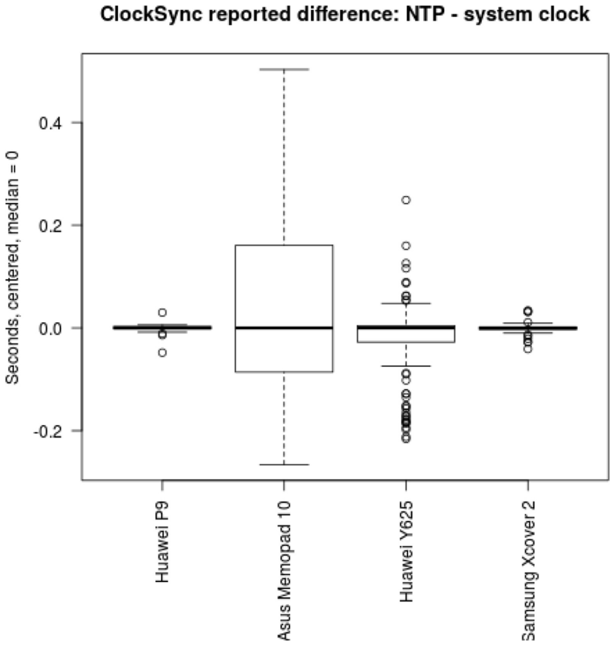


Fig. 3. Results from app that compares system time with NTP time. Results are centered to median = 0.

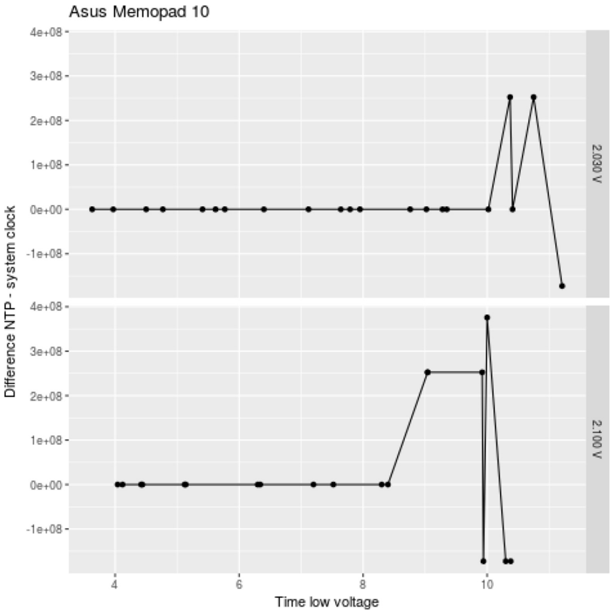


Fig. 4. Asus Memopad 10, clock at voltages 2.030 V and 2.100 V.

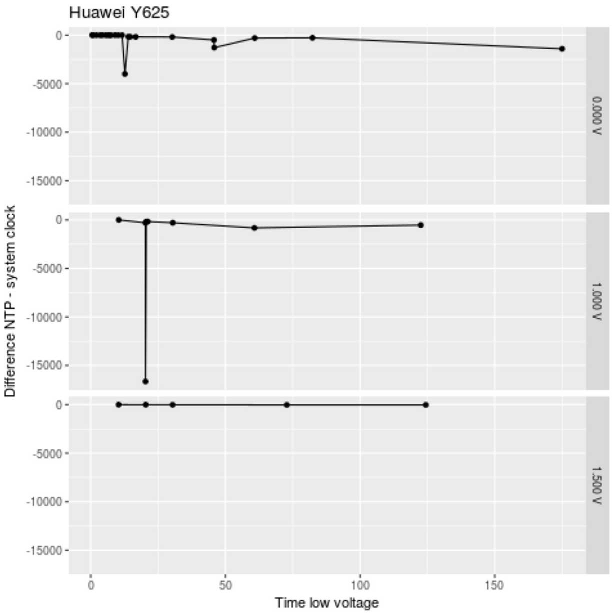


Fig. 5. Huawei Y625, low voltage state.

adjusted forward in time to 2025-07-20, 252460800 and 252460816 s forward in time, before it is reset to 1970-01-02 00:00:00Z. When measuring at 2.100 V, one observation showed the clock to be set to 2029-06-18 19:07:35, or 375831104 s forward in time.

To analyze this from equation (7) and Fig. 1(d), we see that this jump either have to be because of a delay, D_X, that would last for the same amount of time, a huge noise source, or a change in one of the clock offsets, d. As we are not still waiting for the results, and observation of other clocks together with the tests of the comparison methods indicated that the Memopad's clock was showing erroneous time, it is safe to presume that it was the clock offset d_S that were changed.

About 1 min after the clock had been adjusted forward in time, it was reset to normal time even though the tablet was set up to not adjust time automatically. After some log hunting, the Google Checkin service showed up as the culprit. If it receives a SSL error because the clock is outside the certificate validity period, Android will automatically set the time from the Checkin server.

The result on the graph before about 9 s, shows that the clock was gradually slowing by a second per second, this might be a sign that the clock actually stopped, but was not reset. This is a small variation that is not visible in Fig. 4.

Test of Huawei Y625-U21 mobile phone

The next device under test was a Huawei Y625-U21 mobile phone, with a Qualcomm MSM8212, 1.2 GHz processor. It was running Android 4.2.2, with all updated on the time of test.

The first test was to see whether a battery removal would reset time. The voltage was set to 0 for a certain time, and the phone powered back up. This means that the longer the phone is powered off, or held at a voltage lower than the needed for operations, the bigger the difference between the system clock and civil time.

The results show that the clock lags a few seconds until the power has been removed for about 12 s. For a period without power longer than this, the clock is reset to a time just before the device shuts down, as shown in Fig. 5. No forward adjustment of time was observed with this device. The clock is never reset to an epoch, but it likely that it stores a timestamp regularly that it will reset to. The big dips in the figure indicates where the device was shut down for a while between tests.

Test of Samsung Xcover 2

The device tested was a Samsung Xcover 2, GT-S7710. It is running a Samsung ARMv7 processor and Android 4.1.2.

The device was tested with various voltages and periods for the low power state. Several variations of voltages and periods during low voltages were tried. Fig. 6 shows the plot for these. The device either is close to correct (within 1 s), or was reset to 2012-01-01 00:00:00 UTC. Apart from the reset, the clock did not slow down, nor did it jump in time.

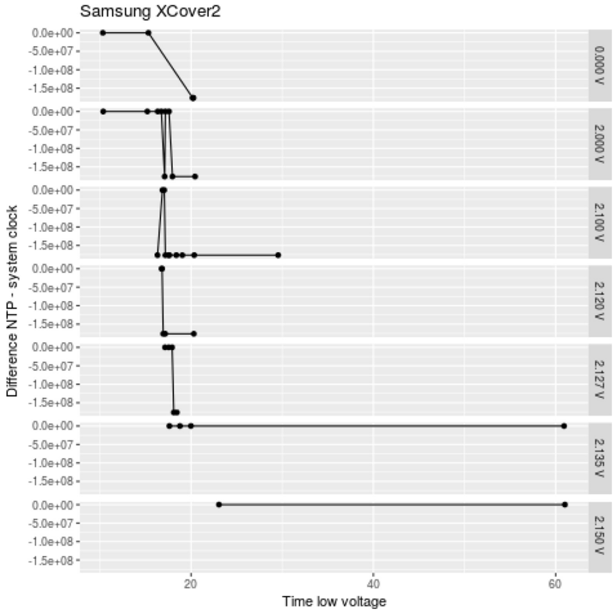
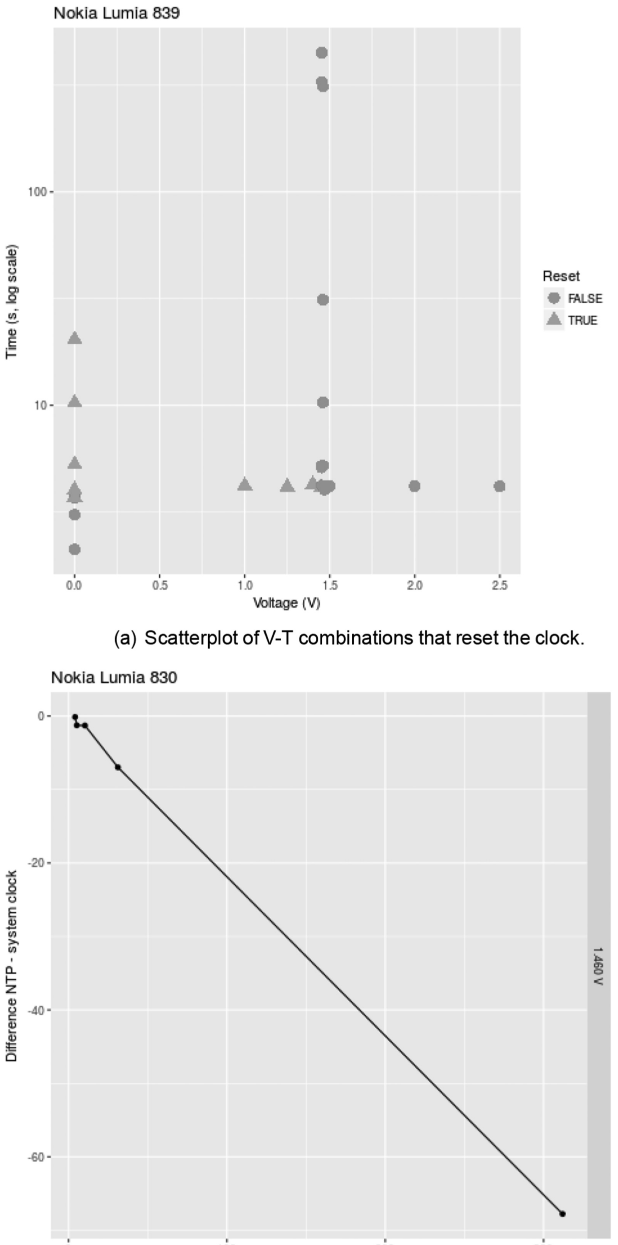


Fig. 6. Samsung XCover 2, low voltage state.



(a) Scatterplot of V-T combinations that reset the clock.

(b) V = 1.460 V.

Fig. 7. Nokia Lumia 830, low voltage state.

As long as the voltage was below 2.135 V, it would reset after between 16 and 17 s, independent on the voltage.

Test of Nokia Lumia 830

The device tested was a Nokia Lumia 830, RM-984, running Windows Phone.

To be able to check the system clock without an app to compare the system clock with a NTP server, the camera was used to take photo of a PC running a NTP client and the timestamps in the EXIF information from the photo with the computer's clock shown in the same photo. This was a more labor-intensive task, and less measurements were therefore taken. The epoch for the device was 2015-04-07 00:00:00 UTC.

Fig. 7 shows a scatterplot of the Voltage–Time combinations that reset the clock, and a graph for voltage held at 1.460 V, while varying time in a low voltage state. The first plot shows the values where the phone is reset, and where it continues to run. Held low longer than 4 s and with less than 1.460 V would reset the device's clock. The difference between the system clock and the NTP clock at 1.460 V shows an increasing difference, but no jumps further than the time it had been under low voltage, indicating that the clock is running slower. 1.455 V and 1.453 V showed the same effect. For all other voltages, the clock was either close to correct, or reset.

The device did not adjust the clock forward in time, but the clock did run slower at about 1.45 V when the low power state was held for 5 min.

Discussion

The results in clock comparisons show that to simply take a picture for documenting time is well within an adequate precision for most forensic examinations. In certain cases where sub-second precision for the comparison is important, one way to reach this can be to juxtapose a correct clock with a stopwatch in a photo, and then compare to the time on the device with the stopwatch in another photo. The stopwatch was the most precise way of measuring with only 0.2 s difference between the extreme values in the comparison with an NTP clock in 268 samples. This research did not look into the drift, or wander of the stop watch, so it might be more imprecise over more time than 5 min.

We did not compare the clocks with a radio controlled clock, controlled by, e.g., the DCF77 signal. The NTP connected clocks were assumed to have adequate precision for this comparison. The accuracy of the clocks was not a concern in this work, as all offset differences were centered to a median of 0.

To use an app on a phone to document the time is a method that in most cases can't be considered “forensically sound”, or “adhering to established digital forensic principles, standards and processes” (Årnes et al., 2017), and should not be used in a forensic investigation. It was used in this paper however instead of other, more time consuming methods for documenting the time, as the other information on the device was not of interest during these tests. Keeping the device connected to the internet could potentially adjust the clock, something that was observed when the clock was set too far into the future, but only after about 30–60 s after boot when the Checkin service contacted the server.

When it comes to the correctness of the clocks of the devices, one of the devices was jumping 8 years into the future, while another was adjusted back in time. Table 4 shows the results for each of the phones tested. The reset time of the Huawei Y625 phone was to a point in time right before the phone was set to a low voltage. The Nokia phone also showed a pattern of slowing down or stopping before being reset.

The Samsung XCover phone showed a pattern fully in compliance with the alternative hypothesis, with no jumps in time or slowing clock before being reset to epoch.

Table 4
Summary of results. “Slow” shows if the clock was lagging, but not reset, after test.

Model	Slows	Result
Memopad	Yes	Jumping 8–12 years forward in time before reset at $V = 2.030$ V and 2.100 V, and $T > 9$ s
Y625	Yes	Reset to a time right before OS shuts down after $T > 12$ s, for $V \leq 1.000$ V
XCover	No	Reset to epoch on $V \leq 2.127$V and $T \approx 18$s
Lumia	Yes	Reset to epoch for $V \leq 1.45$ V and $T \geq 3.72$ s

This has implications for the forensic examination of the timestamps. As we have shown, one of the tested devices could refute our alternative hypothesis. From the limited sample size, we can't conclude with how likely it is for a device to have such behavior, but we can conclude that it is possible and should be taken into consideration that clocks might behave unexpectedly under low voltage states.

In this paper, we did not consider what happens if a device, such as a phone, is set to automatically adjust the time from the network. Just as the automatic clock adjustment happened when the SSL certificate error happened, we can assume that a device will be corrected quickly after a clock error happens. For smaller, resource constrained, and autonomous operating devices, as promised by IoT, the probability for errors that affect the clock can also increase.

Conclusion and future work

It is basic knowledge for a digital forensic examiner to document the state of the devices, including the clock settings. This work has shown that the precision of various ways to document the clocks is within a second, and well within the limits a forensic examination demands. If a precision of less than 0.2 s is needed, a photo together with a stopwatch synchronized with a correct clock can be used to verify the clock settings.

For devices running clocks, when the voltage is low enough, the normal operation of the processor will stop, and only the clock will run. At a certain point, the clock is reset. For one of the devices, the Asus Memopad 10, the clock jumped 8 and 12 years into the future. This is not the clock increasing or decreasing in frequency, but seems to be an error in the memory location storing the current value of the clock.

The implication of this is that it is a possibility for the clock to jump forward in time when the operating system is not running. The test was done by adjusting the voltage on the battery connections to mimic a battery with low voltage, such that this is a state that might happen during normal operations of the device.

This work has not uncovered the exact source of the errors, but rather established that unexpected error effects do appear when the device is close to its operating limits.

For future work, the delay model can be refined to more fine grained model, which take into account all internal calls and atomic operations in the software, all sources of noise in the model, memory and addressing errors, and other external sources of error, to get a more exact model of the possible errors and delays. More devices can be tested, and they can be tested for other hardware or software errors that can happen during its lifetime.

Furthermore, for a forensic investigator a method or set of methods to test and detect violations of such assumptions can improve the quality of the digital forensic field. One thing is to detect violations of the order of dependent events or absurd clock values, but how to find and validate the open and hidden assumptions that is the foundation investigators hypotheses would be a future work.

Acknowledgement

The research leading to these results has received funding from the Research Council of Norway program IKTPLUSS, under the R&D project “Ars Forensica – Computational Forensics for Large-scale Fraud Detection, Crime Investigation & Prevention”, grant agreement 248094/O70.

We would like to thank Katrin Franke for discussions and suggestions to this paper, and Habtamu Abie for good discussions.

We would also thank the National Criminal Investigation Service for the use of lab facilities.

References

Årnes, A., Flaglien, A., Sunde, I.M., Dilijonaite, A., Hamm, J., Sandvik, J.-P., Bjelland, P., Franke, K., Axelsson, S., 2017. Digital Forensics. John Wiley & Sons, Ltd.

Barenghi, A., Bertoni, G., Parrinello, E., Pelosi, G., 2009. Low voltage fault attacks on the RSA cryptosystem. Fault diagnosis and tolerance in cryptography. In: Proceedings of the 6th International Workshop, FDTC 2009, pp. 23–31.

Boyd, C., Forster, P., 2004. Time and date issues in forensic computing - a case study. Digit. Invest. 1 (1), 18–23.

Casey, E., 2002. Error, uncertainty, and loss in digital evidence. Int. J. Digit. Evid. 1 (2), 45.

Dreslinski, R.G., Wieckowski, M., Blaauw, D., Sylvester, D., Mudge, T., 2010. Near-threshold computing: reclaiming moore's law through energy efficient integrated circuits. Proc. IEEE 98 (2), 253–266.

Erbacher, R.F., 2010. Validation for digital forensics. In: ITNG2010–7th International Conference on Information Technology: New Generations, 756–761.

Kaart, M., Laraghy, S., 2014. Android forensics: interpretation of timestamps. Digit. Invest. 11 (3), 234–248. https://doi.org/10.1016/j.diin.2014.05.001.

Martin, J., Burbank, J., Kasch, W., Mills, P.D.L., Jun. 2010. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905. https://rfc-editor.org/rfc/rfc5905.txt.

Qualcomm Technologies, 2016. Qualcomm® Snapdragon™600 Processor APQ8064 Data Sheet. Tech. Rep.

Santos, H., Semiao, J., Cabral, R., Romao, A., Santos, M.B., Teixeira, I.C., Teixeira, J.P., 2016. Aging and performance sensor for SRAM. In: 2016 Conference on Design of Circuits and Integrated Systems, DCIS 2016-Proceedings, pp. 63–68.

Tang, A., Sethumadhavan, S., Stolfo, S., 2017. {CLKSCREW}: exposing the perils of security-oblivious energy management. In: 26th {USENIX} Security Symposium ({USENIX} Security 17). {USENIX} Association, pp. 1057–1074. Vancouver, BC. https://www.usenix.org/conference/usenixsecurity17/Ωtechnical-sessions/presentation/tang.

Willassen, S., 2008a. Hypothesis-based investigation of digital timestamps. IFIP (Int. Fed. Inf. Process) Med. Inf. Monogr. Ser. 285, 75–86.

Willassen, S.Y., 2008b. Finding evidence of antedating in digital investigations. In: ARES 2008–3rd International Conference on Availability, Security, and Reliability, Proceedings, 26–32.

Zhou, H., Nicholls, C., November 2008. Frequency Accuracy & Stability Dependencies of Crystal Oscillators. Tech. Rep.. Carleton University http://kunz-pc.sce.carleton.ca/thesis/CrystalOscillators.pdf.