GTA: GSMem Two-factor Authentication

Andrew Kwong
University of Michigan

Connor Bolton
University of Michigan

Todd Austin
University of Michigan

Abstract

Two-Factor authentication(2FA) is used to keep online accounts protected even in the event of a compromised password. Despite offering stronger security, two-factor authentication systems are still yet to see widespread adoption due to user end inconvenience. Such systems often require the user to interact with a physical token or a phone, and then type a code into a browser; as such, even when it is offered, most users opt to turn off two-factor authentication for convenience's sake.

To address this issue, many recent papers have proposed alternative solutions that require little to no user interaction. Each proposed system, however, has its own drawbacks; they are often incompatible with commonly used phones, computers, and browsers, or require additional hardware. In this paper, we propose leveraging the results of a recent paper, GSMem [1], to create a proof of concept two-factor authentication system that requires absolutely no user interaction; the only requirements are a mobile phone, and the ubiquitous Streaming SIMD Extensions 2 (SSE2) instruction set.

To prove the viability of our proposed system, we built a prototype called GSMem Two-factor Authentication(GTA). By using the electro-magnetic radiation(EMR) emitted by a machine at GSM frequencies to transmit a code to the user's phone, GTA can ensure that the supplicant is within close range to the user, thereby making physical proximity the second factor.

1 Introduction

Databases containing site passwords are routinely stolen, and the hashes contained therein are subsequently cracked. Hackers then indiscriminately compromise the accounts corresponding to every hash that they crack. By requiring a second factor for authentication, in addition to the password, a user can protect him or herself almost entirely from such attacks. Now, only a determined attacker that is targeting the user specifically will be able to steal and use the second authentication factor.

Currently popular two-factor authentication systems include security questions, security tokens, and interaction with the user's phone. The primary issue with such 2FA systems, however, is that each requires additional effort on behalf of the user each time he or she logs in. Furthermore, the answers to security questions can be forgotten, while physicaly security tokens are easily lost. Unfortunately, studies [3] have shown that requiring any slight amount of user interactivity is enough to deter users from using 2FA.

Researchers thus recognized that a user interaction free 2FA system is highly desirable. Bluetooth, NFC, and other sensors based 2FA systems have been proposed [4] that aim to use the supplicant's proximity to some object the user posesses the second factor. All solutions thusfar are limited by the fact that they either require additional hardware support, or are incompatible with modern browsers.

One particularly promising solution, called Soundproof [2], aimed to use sound as a proxy for locality. They took sound samples from both the supplicant's microphone and the user's phone's microphone, and then compared them to see if they were colocated. Their approach, however, left a few things to be desired. We found that the similarity score threshholds varied too dramatically between different pairs of microphones; this resulted in unnacceptably high false positive and false negative rates when a wide variety of microphones were in use. Figure 1 illustrated this failing. Furthermore, not all machines have microphones. Desktops, in particular, are one class of machines unlikely to be able to make use of Soundproof.

To address the shortcomings of current and proposed 2FA systems, we developed GTA, a proof of concept for a 2FA system that requires no user interaction, and doesn't make use of any hardware that isn't ubiquitous or close to it.

Mic Pair	False Positive	False Negative
1 and 2	0.51%	0.51%
1 and 3	0.51%	0.51%
1 and 4	0.00%	0.00%
2 and 3	3.57%	3.06%
2 and 4	2.04%	2.04%
3 and 4	1.02%	1.02%

Figure 1: The first column contains all combinations of pairs taken from 4 distinct microphones

At a high level, GTA works as follows: when a user wishes to log in to a web service, he provides his password as the first factor. If it is correct, the server then sends a randomly generated code to the supplicant. The supplicant then uses the techniques of GSMem [1], which are described in further detail in the next section, to transmit the code to the user's nearby phone. Finally, the code is encrypted with a pre-shared key, and then relayed back to the server, which grants access if and only if the received code matches the original, randomly generated code. Thus, the supplicant must be close to the user's phone to transmit the secret code in order to obtain the second factor, as desired.

For the duration of this paper, the "transmitter" refers to the supplicant's machine, which is responsible for transmitting the code to the user's phone. The "receiver" is the phone that receives said code. Unfortunately for researchers investigating the internal workings of baseband chips, the baseband industry is extremely secretive and refuses to release any source code. Because of this, we opted to use an older feature phone that the open-source community has reverse engineered and developed baseband firmware for; as such, it is highly likely that modern phones can be used as the "receiver" much more effectively.

The rest of this paper is organized as follows: Sections 2 and 3 will describe the implementations of the transmitter and receiver, respectively. Section 4 will describe the experimental setup. Section 5 will evaluate our results. Section 6 concludes, and section 7 details our group dynamics.

2 Transmitter

To implement the transmitter, we borrowed the techniques of GSMem [1], and used the computer's memory bus as an unintentional antenna. When data travels across the wires, EMR is emitted at approximately the frequency of the bus's clock (800MHz for our setup). Since most memory bus clocks operate at a similar frequency, the radiation falls out across the GSM-850, UMTS-850, and LTE-850 bands, and can thus be received and demodulated by a nearby phone's baseband. We made use of Intel's Streaming SIMD Extensions 2(SSE2) instruction set to maximize this effect. In particular, we used the _mm_stream_si128 intrinsic, which corresponds to the MOVNTDQ (move non-temporal double quadword) SSE2 instruction; this instruction moves 128 bits from memory to the CPU's SSE registers, while ignoring caches. The non-temporality of the instruction is essential, as it is the physical transfer of data that is responsible for emitting electromagnetic waves.

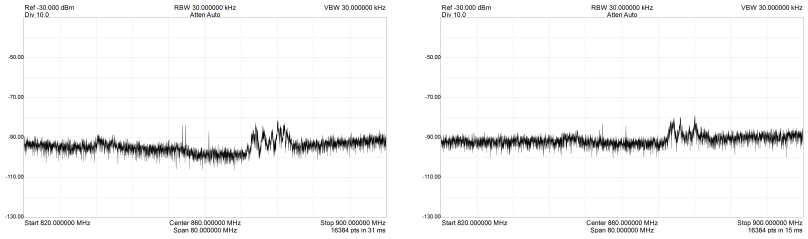
If we repeatedly execute this instruction in a loop, the result is an increase in power around the 800MHz band. Figure 1 compares the EMR from regular activity to the EMR measured while the machine is transmitting. Upon visual inspection of the radiation's fallout among the GSM-850 band, we decided that 860MHz is a good carrier frequency for modulating our signal at, given the nearly 10 dBm difference. The pseudo-code below outlines how we transmit arbitrary binary. In particular, the gsmem_transmit8() function transmits 8 bits at a time.

```
void gsmem_transmit8(data) {
    uint8_t;
    buffer[164096];
    uint8_t *buf_ptr;
    uint8_t t=2000;
    for bit in data:
        if bit==1:
            start=get_current_time();
            buf_ptr=buffer;
            while (t>(get_current_time()-start)):
                _mm_stream_si128(buf_ptr, SSE register);
                buf_ptr+=16;
        else:
            sleep(t);
}
```

Building upon this phenomena, we modulate arbitrary binary using the binary amplitude shift keying (B-ASK) scheme. To transmit a 1, we fully utilize the memory bus for t seconds; to transmit a 0, we do nothing for t seconds. In figure 2 we can observe how doing so results in a nearly 10dBm difference at 860MHz.

3 Receiver

Due to the baseband industry's largely successful efforts at security through obscurity, we resorted to using the Motorola C118 feature phone for our receiver; this is be-



(a) Frequency domain from casual emissions. (b) Frequency domain while transmitting high.

Figure 2: Power measurements taken just above the laptop

cause the OsmocomBB(Open-Source MObile COMmunication BaseBand) project has developed the only open source GSM baseband implementation, for certain compatible phones. As such, osmocombb was the only way by which we could feasibly modify the source code for a baseband chip.

The baseband firmware was modified so that the RTOS's main event loop calls an additional function with each iteration. It is this additional function that handles the demodulation of the code.

If samples are taken at a higher rate, the receiver can be made more resilient to noise; due to limitations in the processing power of such an old feature phone, however, we found that using t=2 seconds and sampling 8 times per period resulted in a reliable channel with a throughput of 0.5 bits/second.

The pseudo-code below illustrates how we modified the C123's firmware.

```
//initialize phone
//set up timers
while(True):
    execute_layer1_tasks();
    update_osmocom_timers();
    handle_keypad();
    handle_l1_l23_interface();
    handle_power_measurements();
    handle_synchronization();
    handle_tone();
    GSMem_receive();
```

In GSMem_receiver, the phone takes power measurements at 860MHz and stores the values in a ring buffer. Once averages for transmission at B-ASK 1 and B-ASK 0 levels are found, the receiver can then demodulate the signal by sampling the DSP. The values are compared against the expected values obtained from transmitting and casual emissions.

4 Experimental Setup

Our receiver can receive the code transmitted through GSMem [1] when the phone is sitting right on top of the transmitting laptop, as shown in Figure 3.

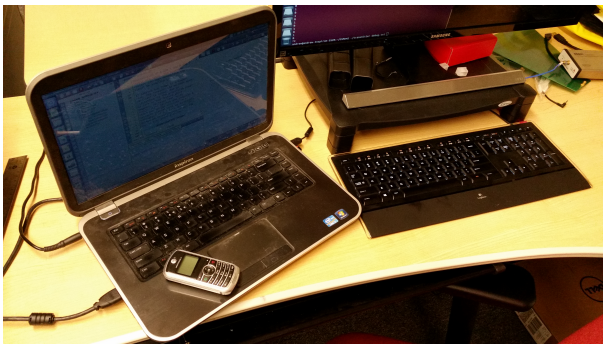


Figure 3: Experimental setup.

The laptop used is an Inspiron 15R 5520, which has DDR3 memory with a bus clock of 800MHz. The phone is the Motorola C118 phone, which has a Calypso chipset.

All experiments were conducted in a graduate student office, exposed to multiple sources of noise and background radiation. There were numerous desktop machines in use within a 5m radius.

5 Evaluation

In this section we will evaluate whether or not the covert channel proposed in [1] can indeed be used to intentionally transmit the code to the receiver. In particular, we will examine the channel's range and throughput.

Guri's [1] study claims that they achieved power level differences of 1 dBm at ranges of up to 110cm. As figure 4 demonstrates, we were unable to achieve the same range as Guri et al. At ranges of just over 4 cm, the lines already converge, and there is no discernible signal.

We attribute this discrepancy to one major factor: shielding. In contrast to the Guri et al., who transmitted from various workstations, we only transmitted from the Inspiron 15R 5520 laptop. While some of their cases had sides made primarily of plastic, doing little to attenuate EMR, our laptop was likely shielded by a metal case,

and we were simply observing EMR leakage from the keyboard. This hypothesis is supported by our observation that the strength of the signal varied dramatically as the phone was moved to different locations on top of the laptop.

This, of course, is problematic; requiring such close proximities may be inconvenient enough that users will not use our proposed 2FA system. Ideally, the phone never has to leave the user's pocket or purse. As such, this 2FA scheme may be more suitable for use on desktops, which can likely transmit stronger signals. Desktops also happen to be the class of machines least likely to be able to make use of Soundproof [2]. Perhaps, then, a combination of both Soundproof and GTA can provide sufficient coverage for commonly used devices.

Regarding throughput, at very close proximities we were able to produce comparable results. While Guri et al. were able to transmit up to 2 bits/second, we were able to achieve a throughput of 0.5 bits/second. There are a few factors that are likely culprits.

For one, we used the Motorola C118, as opposed to the C123, to implement our receiver. The C118 is an older model, with reduced computational power, and possibly a less sensitive/accurate DSP. Another is that the weakened signal compounds that fact that the channel becomes more susceptible to noise at higher bit rates; as the period shortens, shorter durations of interference can flip bits. It also quite possible that Guri et al. simply implemented more effective noise filtering in the phone's baseband.

The low throughput of the channel also poses a problem. In order for a code to be secure, it must be of sufficient length. By using an exponential back-off scheme, where a supplicant must wait for intervals of increasing length between attempts to log in with incorrect codes, we can reduce the required length of the code. However, users will not want to wait longer than a few seconds at most, and as such, a higher throughput channel is required. We believe that use of a modern phone can remedy this with its improved processing power and baseband sensitivity.

6 Conclusion

In this paper we described a two-factor authentication system wherein physical proximity to the user's phone is the second factor; the supplicant is granted access only if he is close enough to transmit a code to the phone. To accomplish this, we modified a phone's firmware, and successfully transfered files via the GSMem covert channel to the phone. Though the signal to noise ratio was only sufficient when the phone was lying directly on top of the laptop, we deem this a satisfactory proof of concept that verifies the feasability of using GSMem to provide locality as a second factor. We leave it to future work to use a modern phone to correct the shortfallings of our low throughput, short-ranged channel.

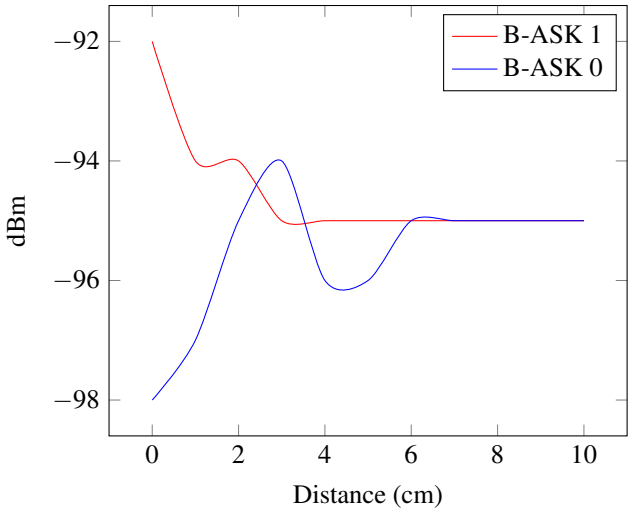


Figure 4: Received signal strength at 860MHz as a function of distance

7 Group Dynamics

Andrew Kwong was responsible for creating both the transmitter on x86 machines, and the receiver on the Motorola C123. He also wrote the client and server for GTA. Connor Bolton reproduced Soundproof's [2] results and found their proposed 2FA system to be less than ideal.

References

[1] Guri, M., Kachlon, A., Hasson, O., Kedma, G., Mirsky, Y., and Elovici, Y. Gsmem; data exfiltration from air-gapped computers over gsm frequencies. In *Proceedings of Usenix Security Symposium 2015* (Aug. 2015).

[2] Karapanos, Nikolaos, e. a. Sound-proof: Usable two-factor authentication based on ambient sound. In *24th USENIX Security Symposium* (2015).

[3] Petsas, T., Tsirantonakis, G., Athanasopoulos, E., and Ioannidis, S. Two-factor authentication: Is the world ready? In *8th European Workshop on System Security* (2015).

[4] Rubenking, N. J. Universal two-factor authentication just got more universal. `http://www.pcmag.com/article2/0,2817,2486835,00.asp`.