zkTAP: A Zero-Knowledge Trustless Authentication Protocol

By

Majdi Hassan

Joseph Kuo

Lilan Yang

Project Proposal for ECE 445, Senior Design, Fall 2019

TA: Evan Widloski

19 September 2019

Project No. 33

1 Introduction

1.1 Objective

How much do you value your credit card? Well it turns out that average credit card is worth about $10 on the dark web. We presume that you value your data much more than a measly $10. Unfortunately the security that exists on RFID credit cards and other RFID authentication schemes is almost non-existent [1]. As a result, these cards are super cheap to make however leave you vulnerable to skimming attacks.

We have created a product for those who do value their data. Rather than transmit your data, we instead transmit proofs that you are the owner. Doing so keeps your data safe as we never transmit your data. And so we propose a Zero-Knowledge RFID authentication system based of the Schnorr Protocol.

1.2 Background

1.2.1 Zero-Knowledge and the Two Balls and Color Blind Friend Example

Imagine that your friend is red-green color blind and you aren't color blind. Your friend has two balls, one green and one red and thinks that the balls are identical. The balls are of course distinguishable and you want to prove to him that they are distinguishable. In order to do so, your friend will leave the room and will shuffle the balls around. Now if you are actually not colorblind, then you will be able to distinguish between the two balls. The process is repeated until your friend is thoroughly convinced that the two balls are different. This is zero knowledge in the sense that your friend gains no knowledge on which of the balls is green/red. Furthermore, he still has no idea how to tell the red and green apart [2].

At an extremely high level, this is how our protocol works. The verifier, even in the dishonest case, learns nothing of your secret key, only that you are the own of the secret key.

1.2.2 Discrete Log Assumption

One assumption that we make is that computing the Discrete Log in a finite field is difficult. Mathematically, this can be represented as, given $A = g^a$, that a is difficult to compute. When we say something is 'difficult to compute,' we mean that there exists no polynomial time algorithm to compute a.
To be complete, g is a generator of the cyclic group G, under multiplication of a finite field of a prime power [3]. It isn't important to understand the sentence above, but we would like to stress that given $A = g^a$, that a is difficult to compute.

1.2.3 Schnorr Protocol

Zero Knowledge Proofs such as the Schnorr protocol has seen wide adoption in the cryptocurrency space. The interesting part is that the Schnorr protocol was actually designed for use in 'smart cards' [4]. It was thought that the Schnorr protocol was too impractical as it needs a reliable random generator and that computational costs were simply too high.
Recently (as in 2016) a robust and cheap number (on the order of $2 $\sim$ $3) generator was discovered by B. Lampert [5]. Based on our calculations (found in design section), we estimate that with a 16 bit microcontroller, we would be able to compute all the cryptographic primitives in around 3 seconds without any precomputing.

1.2.4 Elliptic Curve Cryptography

Elliptic Curve Cryptography is a public key encryption scheme based off of the algebraic structure of elliptic curves (In particular adding and multiplying) and that the assumption that discrete log is difficult to find. As a result, it requires less bits than RSA and DSA which both rely on the assumption that factoring a number made up of two large prime numbers is difficult (since an attack would involve an attack on one of the prime numbers). Furthermore Bernstein discovered several optimization on elliptic curve operations over prime numbers that can represented by a non power of 2 number of bits (ie 255)[6].
As a result we will use Curve22519 with the prime number $2^{255} - 19$. It is under public domain and it is approved for use by the US Federal Government [7].

1.2.5 Avalanche Breakdown of a Zener Diode

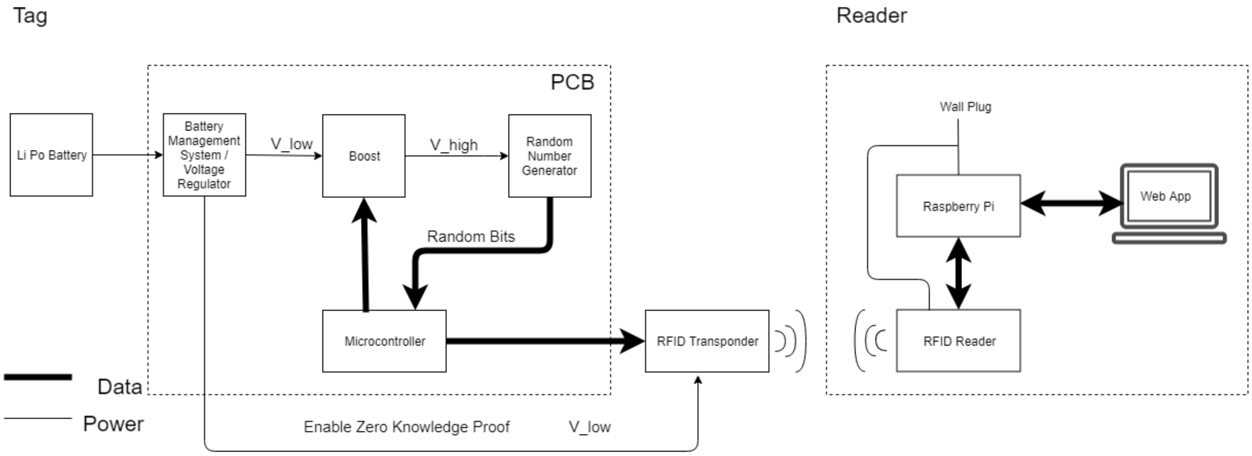
Breakdown occurs when the current experiences a large increase in a short amount of time while under reverse bias. The Avalanche Breakdown of a Zener Diode occurs when the electron-hole pairs go through impact ionization which is the process of a charge carrier losing energy due to the creation of other charges. We can use the randomness of these collisions as an entropy source [5].

1.3 High-Level Requirements

- The response given by the tag must be accepted by the reader with all but negligible probability.
- A user must be authenticated in under 10 seconds.
- Given interaction with a dishonest verifier (ie when someone tries to skim you), the dishonest verifier must learn nothing about your secret key. Furthermore an eavesdropper must not be able to impersonate a user (except with negligible probability).

2 Design

2.1 Block Diagram



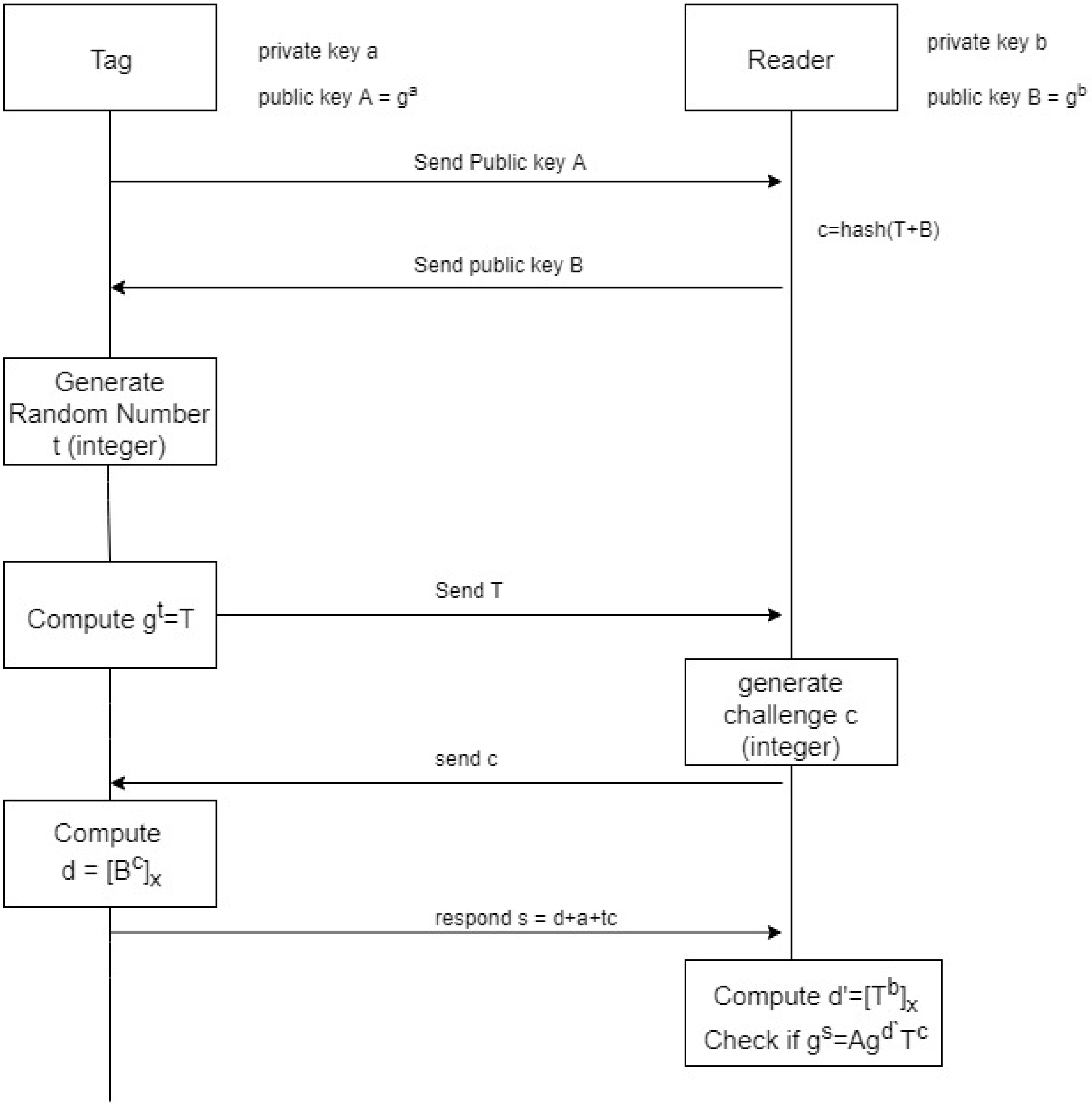
We assert that our protocol satisfies our first requirement. A short and straightforward proof is attached in Appendix A.

For requirement 2, we will make the assumption that the processing power of the reader is at least a magnitude greater than the microcontroller. Thus any computational costs of the reader are therefore negligible. Thus we can reduce our problem into: generation of a 255 bit random number, 2 Elliptic Curve Multiplications, 1 addition (mod p). An elliptic curve multiplication on Curve22515 takes $12.34 * 10^6$ clock cycles on MSP430 microcontroller [8]. A MSP430 micrcontroller can run between 8 MHz-24 MHz depending on the generator of the microcontroller [9]. Lampert circuits can be safely sampled at a rate of around $10^4 \sim 10^6$ Hz [5]. Thus we can expect that the random number generation portion takes roughly 0.03 seconds. In addition, we will assume that addition take negligible time. We will assume the worst case scenario which would imply that our protocol would take about 3-4 seconds. We could likely get this down to 1 Elliptic Curve multiplication if we allow for precomputing. There are two ways to further speed this up, we could a 32 microcontroller or use a FPGA + microcontroller combination.

Requirement 3 can only be realized if we have a random number generator on the tag. Consider the dishonest reader example. The tag's response is $s = d + a + tc$. If a reader were able to predict the value of t, then that reader would be able to extract the secret key a, as s, d, t and c are all known to the verifier. By introducing a random number generator, we can maintain privacy of the tag.

In the case of an honest verifier, an eavesdropper actually can't figure out whether or not authentication was successful, as the value d or d' can only be calculated by the tag and by the reader.

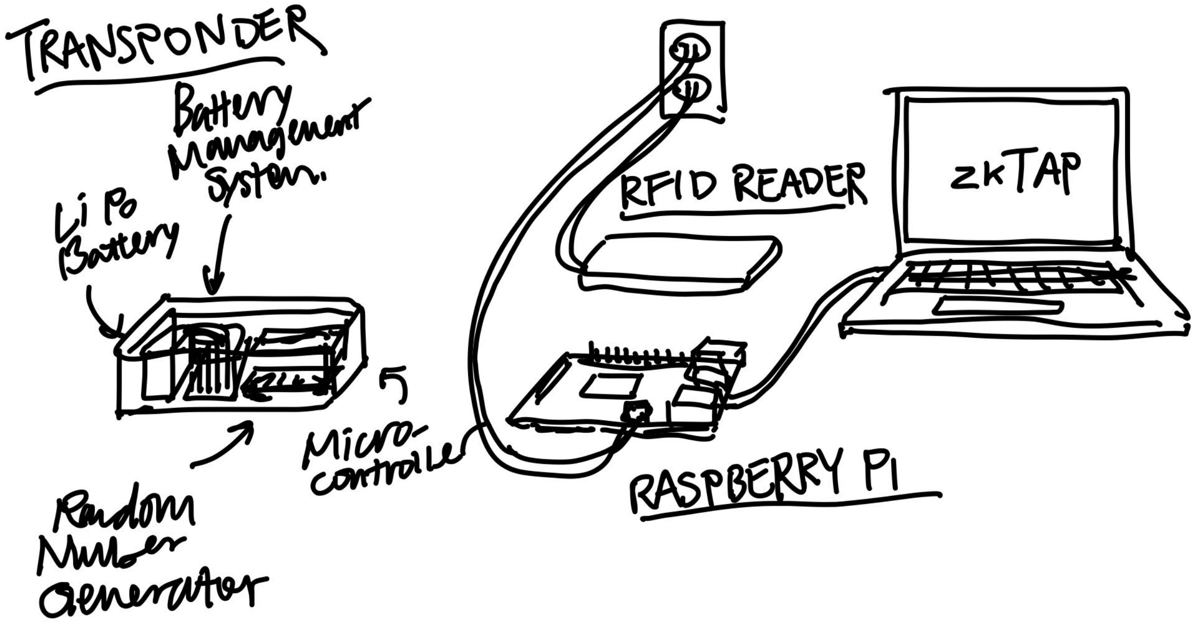
2.2 Output of Cryptography Protocol



This follows several main steps. Our protocol is based off of the Schnorr protocol outline here [10].

1. Exchange public keys.
2. Tag generates t, a blinding factor and sends a commitment to the reader.
3. The reader generates a challenge.
4. Tag responses to the challenge.
5. Reader verifies that the challenge is correct.

2.3 Physical Design



2.4 Functional Overview

2.4.1 Li Po Battery

The Lithium polymer battery powers our tag. We anticipate that our random number generator will consume the most amount of power as we will have to supply around 12.5V to the Zener diodes.
Requirement: The lithium polymer battery must store enough charge to power our random number generator for at least a minute. This is sufficient as this would be enough to provide roughly $\sim$ 2000 authentication attempts.

2.4.2 Battery Management System/Voltage Regulator

Given the potential dangers of Lithium batteries, our battery management system prevents are Li Po battery from overcharging so that we don't have to babysit our battery pack when it is charging. Given how sensitive our protocol is our random number generator it must stop supplying power when the voltage supplied falls out of specifications for our boost converter.
Requirement: Our voltage regulator must be able to supply 3.3V +/- 5% from our battery.
Requirement: Our BMS must shut off when we can no longer supply enough power.

2.4.3 Boost

The main purpose of our boost convert is to provide enough voltage to keep our Zener diode in reverse breakdown. Typically this is around 12V, however we would like to keep the diodes at around 12.25-12.5V as a safety buffer.
Requirement: Given a 3.3V source, our boost converter must be able to supply 12.25-12.5V.

2.4.4 Random Number Generator

Our random number generator provides random numbers to be used to create blinding factors. We intend on using the randomness of breakdown current in a Zener diode. This will fed into a 1 bit Analog to Digital Converter and sampled by our microcontroller.
Requirement: Because our project involves RF, the ‘randomness’ of circuit must not be sensitive to RF.

2.4.5 Microcontroller

Our microcontroller is responsible for sampling bits from the Random Number Generator, performing Elliptic Curve operations and sending information to the transponder to be sent to the reader. We will not be implementing our own Elliptic Curve as we would not like to ‘roll in our own crypto.’
Requirements: Must be able to perform an Elliptic Multiple operation over Curve22519 within 1.5 seconds (This is higher our estimate).

2.4.6 RFID Transponder

Given that the software side of our protocol already satisfies ‘the response given by the tag must be accepted by the reader with all but negligible probability,’ our transponder must maintain this and transmit our data free of corruption. Our microcontroller will likely provide an error correcting code.
Requirement: Given that our serializated form a 255 bit int/elliptic curve point + Error Correcting Code will likely exceed the bit width of our transponder, we will likely have to transmit each step of our proof in multiple steps. As a result we would require the total sum of all the steps in our proof to be transmitted in under a second.

2.5 RFID Reader

We would like to use a third party RFID Reader to transmit and receive the proofs from the tag. It enables contactless communications with the tag.
Requirement: It reads to the RFID tag.
Requirement: It has to support Error Correction Codes.

2.6 Raspberry Pi

Our Raspberry Pi is responsible for verifying that the proofs that the tag sent was actually true. In our analysis above, we made the assumption that the verification part was computationally negligible. It also is responsible for interacting with the transmitter and has to host our web application.
Requirement: Elliptic Curve operations are computationally negligible.

2.7 Web Application

We would also create a simple web application as an user interface. Our goal is to better visualize all the functionalities of the tag.
Requirement: The web application would be able to indicate if a user has been successfully authenticated.
Requirement: Additionally the interface would allow to add new users and revoke access.

2.8 Risk Analysis

Since we have created a demonstration our protocol in Python, we don’t think the cryptography portion to be of significant risk to our problem. In addition we will be using a 3rd party implementation of the Elliptic

Curve Cryptography primitives as such implementations are sensitive to various side channel attacks.
The RFID Transponder/Reader portion represents significant risk to our project as it is very difficult to do a RFID project without a RFID Transponder or Reader. Despite this, we don't actually anticipate this being a significant problem.

The random number generator portion of our project represents a medium amount of risk. It might actually be difficult to pass a third party testing suite for a random number generator. However the upside is that if this circuit doesn't end up working, we could use boot time to seed a software random number generator (This is not secure as boot time isn't random!) and so we could still demonstrate our product if this doesn't work.

The power portion of our project also represents a low-medium amount of risk, as we could use a 3rd party power supply if this part ends up not working.

3 Ethics

In regards to safety, the issue that may arise pertains with our use of a lithium battery. If not properly understood when it comes to its implementation, fires and other accidents become likely risks to occur and would pose danger to ‘the safety, health, and welfare of the public’ [11]. In order to mitigate these risks, proper practices must take place. First, the specifications of the battery must be thoroughly looked over and understood, so when it comes to implementing it in conjunction with the rest of the hardware, there is no overload or improper design such as short circuiting or placing near an area of high temperature that could cause the battery to catch fire and result in a hazardous situation. According to the University of Washington, certain practices such as proper procurement, storage, charging practice, and disposal should be taken into account [12]. For example, acquire the battery from a trusted source, place them away from flammable materials, disconnect batteries that exhibit any unusual behavior such as heating up or releasing some sort of smell, and dispose of batteries safely by taking them to designated facilities.

We plan on using 120∼150 kHz band which is unregulated so we won’t have to worry about infringing on any regulations.

Curve22519 is under public domain and so we don’t have to worry about patent infringement.

We understand the risks and ‘societal implications of conventional and emerging technologies,’ which is why we are creating this product. As a result, our top priority is to protect the user’s private information [11].

References

[1] D. Giese, K. Liu, M. Sun, T. Syed, and L. Zhang, “Security analysis of near-field communication (nfc) payments,” 04 2019.

[2] K. Chalkias, “Demonstrate how zero-knowledge proofs work without using maths,” Sept 2017.

[3] A. M. Odlyzko, “Discrete logarithms in finite fields and their cryptographic significance,” *Proc. Eurocrypt*, no. 84.

[4] C. P. Schnorr, “Efficient identification and signatures for smart cards,” in *Advances in Cryptology — CRYPTO’ 89 Proceedings*, G. Brassard, Ed. New York, NY: Springer New York, 1990, pp. 239–252.

[5] B. Lampert, R. S. Wahby, S. Leonard, and P. Levis, “Robust, low-cost, auditable random number generation for embedded system security,” in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, ser. SenSys ’16. New York, NY, USA: ACM, 2016. [Online]. Available: http://doi.acm.org/10.1145/2994551.2994568 pp. 16–27.

[6] D. J. Bernstein, “Curve25519: New diffie-hellman speed records,” in *Public Key Cryptography - PKC 2006*, M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 207–228.

[7] H. M. T. S. Langley, A., “Elliptic curves for security,” Internet Research Task Force, January 2016.

[8] Z. Liu, J. Großschädl, L. Li, and Q. Xu, “Energy-efficient elliptic curve cryptography for msp430-based wireless sensor nodes,” in *Information Security and Privacy*, J. K. Liu and R. Steinfeld, Eds. Cham: Springer International Publishing, 2016, pp. 94–112.

[9] “Msp430f552x, msp430f551x mixed-signal microcontrollers,” Texas Instruments, Dallas, TX, 2008.

[10] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel, "A new rfid privacy model," in *ESORICS*, 2011.

[11] "IEEE IEEE Code of Ethics," 2019. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[12] "Stay safe when using lithium batteries," April 2018. [Online]. Available: https://www.ehs.washington.edu/about/latest-news/stay-safe-when-using-lithium-batteries

Appendix A Proof of Requirement 1

Requirement 1 can be rewritten as, $g^s = Ag^{d'}T^c$. Now we need to show that this shows true for all users.

Lemma A.1. *The value d and d' are equivalent so long so as either the value t (blinding factor generated by the tag) or the value y (secret key of the reader) is known.*

Proof. Recall that $d = [B^t]_x$ and $d' = [T^b]_x$, where $[]_x$ is the x coordinate of the elliptic curve point represented by the value in brackets. Substituting values in, we get $d = [g^{bt}]_x$ and $d' = [g^{tb}]_x$ and thus we see that the values are equivalent. As we can see knowledge of either t or b is required to construct d. □

Proof. Let s represent the responsive given by the tag. We need to show $g^s = Ag^{d'}T^c$ hold true.

$$\begin{aligned}
g^s &= Ag^{d'}T^c \\
&= (g^a)g^{d'}g^{tc} && \text{(substituting variables)} \\
&= (g^a)g^{d}g^{tc} && \text{(from lemma a.1)} \\
&= g^{a+d+tc} && \text{(the tag's response is s = a + d + t c)} \\
&= g^s
\end{aligned}$$

□