CPSC 502.04: Research Project in Information Security (F21)
Supervisor: Rei Safavi-Naini <rei@ucalgary.ca>

UNIVERSITY OF CALGARY

Name: Kiley Carson UCID: 30009983

Interim Report

Due 12-09-2021 @ 23:59

Introduction

Authentication is the process of verifying the identity of a user and restricting illegitimate users from accessing the system [1]. FIDO2 is the Fast IDentity Online (FIDO) Alliance's proposed standard for passwordless user authentication; the goal was to be both more secure than password with multi-factor authentication (MFA) and easier to use [2]. The standards are compromised of W3C's Web Authentication (WebAuthn) and the FIDO Alliance's Client-to-Authenticator Protocol v2.0 (CTAP2). FIDO2 is a token-based system that this project intends to strengthen via a One-time Password. The projects goals include understanding the FIDO2 authentication system, its cryptographic building blocks, the security model and the proof. Additionally, the deliverables include a design of a security enhancement to the FIDO2 protocol, a proof of concept implementation of the design and an evaluation. The security enhancement is based on the vulnerabilities in the CTAP2 protocol and modifications made in the strong-unforgeability design proposed by Barbosa, *et al* [3].

CTAP2 Vulnerabilities

There is a need for a security enhancement in the Client to Authenticator Protocol (CTAP2). The participants involved in the CTAP2 protocol are the users, the relying-party/clients/browsers and authenticators/tokens. These entities, participate in the three phases, Setup (Figure 1), Binding (Figure 2) and Authentication/Validation (Figure 3), which combined comprise the protocol. The setup procedures first generate an attestation key pair for the authenticator and configure a user PIN by embedding the PIN in the authenticator (Figure 1). Storage of the embedded PIN is assumed to be initialized using a procedure carried out under special setup trust assumptions [3]. An authenticated channel is assumed for the communications between the client and the authenticator during setup as there are no "pre-established authentication parameters". In the authenticator setup phase the user embeds its PIN into the authenticator via a client (browser) and as a result, the authenticator stores a PIN-related long term state. If this is violated the authenticator could store the wrong PIN-related long term state.

CTAP2 is secure in the security model unforgeability with Trusted Binding (UF-t). This means that adversaries cannot forge new authorized messages when the channel between the authenticators and clients have assumptions of trust [3]. However, CTAP2 does not achieve Unforgeability (UF) security. This is because unauthenticated Diffie-Hellman Key Exchange (DHKE) is used between the client and the token in both the Setup and Binding phases, which is vulnerable to Man-in-the-Middle attacks [3] allowing for forgery of authorized messages.

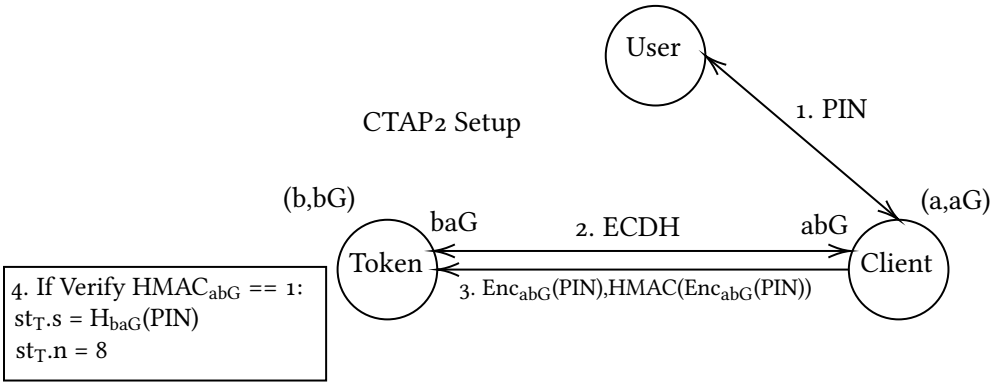

Figure 1: CTAP2 Setup Protocol

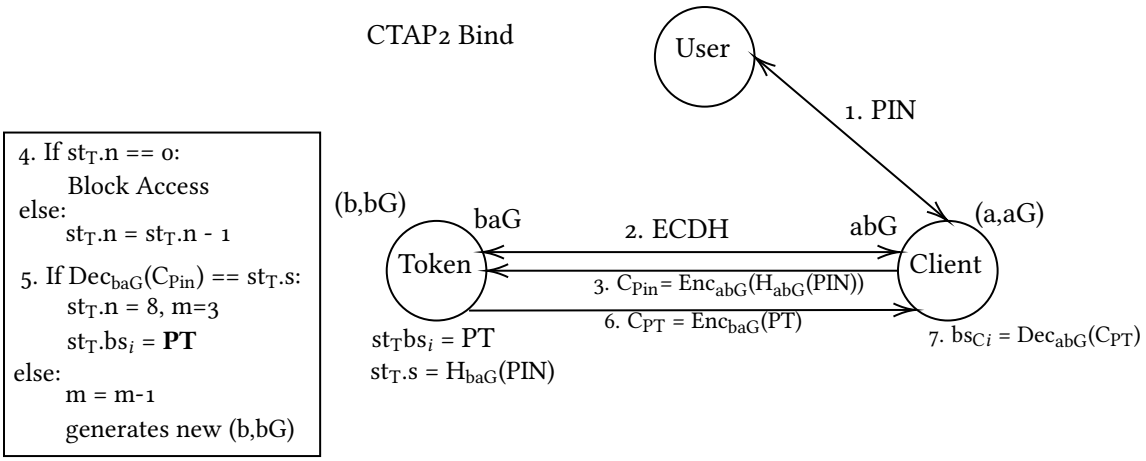

Figure 2: CTAP2 Binding Protocol

The Binding phase occurs after the Setup phase to establish a "session key" that can be used in the third (Figure 2). Another flaw in the CTAP2 protocol is that all access channels between the client and the token, share the same binding state (the pinToken PT). For security, this means that if any of these channels are compromised by an attacker then the remaining clients are no longer secure. To achieve stronger security all the binding states must be independent from each other. Then even if only access channel is not compromised, it is still possible to maintain its security.

Proposed Design

The first thing that was considered for the enhanced design of the CTAP2 protocol was a substitution of a password for a pin. However, after some investigation on the requirements for the CTAP2 PIN it was determined that this change would not enhance the security of the protocol. This is because the current requirements for the PIN are a minimum length of 4 and maximum length of 63 bytes in the UTF-8

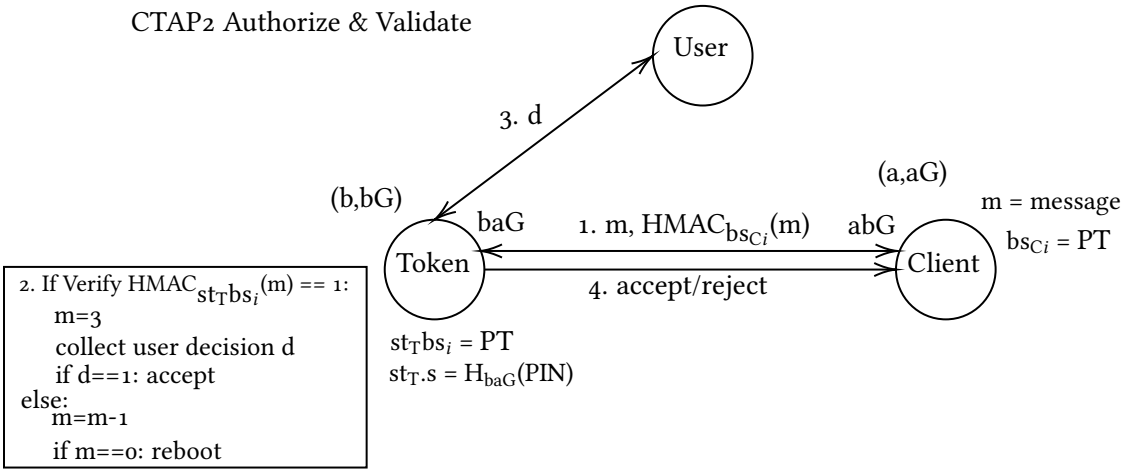

Figure 3: CTAP2 Validation and Authentication Protocol

encoding form [4]. The design will also include the security enhancements from the Strong Unforgeability Improvements that Barbosa, *et al* [3]. One improvement is the switch from using unauthenticated Diffie-Hellman Key Exchange (DHKE) to a password-authenticated key exchange (PAKE) protocol between the client and authenticator in the Setup and Binding phases since PAKE protocols take as input a common password and output the same random session key for both parties [3]. An second design consideration is the addition of user confirmation like the press of a button in the binding phase to prevent malicious binding attempts. This will allow the design to eliminate useless reboots that hamper usability in order to test user presence. There are reboot after 3 consecutive mismatches/errors with messages received by the token from the client.

Additionally, CTAP2 design has to be modified to accept a Time-Based One Time Password (TOTP) seed instead of a PIN in Setup. This allows the binding states to be derived using the TOTP algorithm which is a time-based variant of the OTP algorithm that provides short-lived OTP values, which are desirable for enhanced security. [5] The Google Authenticator specifically uses an offline variety of TOTP where a prior sharing of the secret seed between the client and the server is required. In this case, it is the user's device (vs. the server) that generates the one-time codes, then both parties are expected to generate the same one-time code during a given time window. Something that still has to be determined is the Setup interaction between authenticator and the server, and how the client will get the seed to the server, because with this design; instead authenticator being restricted to a token, there is more flexibility (the authenticator can be a mobile device).

Project Plan

Completion of the project will require focus in two main areas, the design and implementation of the proof of concept. In order to complete the design portion of the project, the following steps need to be completed:

- Confirmation the Time-Based One-Time-Pad will in the design as desired

- The completion of a detailed security analysis
- A computational security proof using the Bellare-Rogaway Model [6]

After the above is completed implementation will be the next main area of focus. This will require:

- The integration of the WebAuthentication and CTAP2 APIs with a client and server application to create the initial FIDO2 set up
- Modifications to the CTAP2 APIs will be required in order to incorporate the Google Authenticator (TOTP) APIs into the system
- Analysis of the implementation of the final system will have to completed after the proof-of-concept is completed

The three APIs mentioned previously are all open-source and available on Github, simplifying the development and implementation process.

References

[1] D. Dasgupta, A. Roy, A. Nag, *et al.*, *Advances in user authentication*. Springer, 2017.

[2] "Fido2: Webauthn & ctap." (2019), [Online]. Available: https://fidoalliance.org/fido2/.

[3] M. Barbosa, A. Boldyreva, S. Chen, and B. Warinschi, "Provable security analysis of fido2," in *Annual International Cryptology Conference*, Springer, 2021, pp. 125–156.

[4] "Client to authenticator protocol (ctap) proposed standard." (2021), [Online]. Available: https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-20210615.html.

[5] "Rfc:6238, totp: Time-based one-time password algorithm." (2011), [Online]. Available: https://www.rfc-editor.org/rfc/rfc6238.

[6] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Annual international cryptology conference*, Springer, 1993, pp. 232–249.