Using blockchain to meet the security requirements of a messaging system identified by the extended STPA method

Júlio César Leitão Albuquerque de Farias, Celso Massaki Hirata
Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos/SP - Brasil

***Abstract*—Blockchain is a growing list of records, called blocks, that are securely linked together using cryptography. Blockchain is employed in different types of applications, such as money transfer, secure sharing of medical data, supply chain and logistics monitoring, voting mechanism, and original content creation. The defense applications have specific security requirements such as integrity, availability, and trust. STPA extended with STRIDE is a method that has been used to identify not only safety requirements but also security requirements. We investigate how blockchain can be used to meet the security requirements, identified by STPA with STRIDE, of a messaging system in a defense organization.**

***Palavras-Chave*—Blockchain, STPA, STRIDE.**

I. Introduction

Blockchain technology started as a means to transfer digital money in a decentralized way. Currently, the cryptocurrency market mobilizes around 1.4 trillion dollars daily, according to *CoinMarketCap*[1]. Blockchain can be seen as a distributed database in which the components of a network can interact to increment it. With this broad perspective, you can use it for other applications. In this way, technology has been developed to allow different forms of transactions and to store different types of content.

There are applications that use blockchain as a database in various sectors, such as auditable supply chain, goods shipping sectors, electronics retail, music industry, investments and loans, as shown in Biswas [1]. Olnes [2] points out that some governments have already started to use blockchain to increase management transparency and speed up public processes through smart contracts.

STPA [3] is a safety analysis method, based on systems theory model STAMP, which was extended to perform cybersecurity threat analyses [5]. STPA allows identifying more loss scenarios due to component interactions than the conventional methods.

The applications of the Ministry of Defense of Brazil have strategic relevance in keeping the Armed Forces ready for employment and in decision-making. Their application needs attributes such as integrity, availability, and trust.

Motivated by the growing use of blockchain technology and the potential benefits of STPA, we investigate how blockchain technology and STPA can be used to secure the messaging system of the Brazilian Ministry of Defense.

In Section II, we introduce STPA and STRIDE which are used to obtain system security requirements and similar works are reviewed. In Section III, we show the proposal to identify benefits of using blockchain to meet system requirements. In Section IV, we implement and test a message exchange system using the proposal. In Section V, we discuss some issues about decentralizing systems; and in Section VI, we conclude our work and give suggestions for future work.

[1]CoinMarketCap is a website that gathers information about cryptocurrency transactions.

II. Theoretical foundation and related work

A. STAMP and STPA

STAMP (System-Theoretic Accident Model and Processes) is an accident causality model based on systems theory, which provides theoretical support for STPA (System-Theoretic Process Analysis). By considering emergent properties of the system, which depart from the relationships between its components, it is possible to go beyond a traditional causality model based on component failure and find unsafe interactions between components that do not have failures but can bring potential damages.

STPA is a method of analysis that analyzes the potential cause of accidents in the development phase so that the risks to the system can be controlled. The STPA analysis has four steps: *Define Purpose of the Analysis* is the step that identifies system losses, hazards, and constraints. *Model the Control Structure* aims to model the hierarchical control structure with the components and their relationships, including control actions and feedback. *Identify Unsafe Control Actions (UCAs)* identifies control actions (CA) of the control structure that are hazardous in a particular context and worst-case environment. UCAs are associated with the identified hazards. *Identify Loss Scenarios* identifies loss scenarios and causal factors of the UCAs. The step also identifies mechanisms to deal with the causal factors.

B. STRIDE

STRIDE [4] is a threat model to identify computer security threats. STRIDE helps to verify the security properties: confidentiality, integrity, and availability, looking for their related threats, respectively: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. STRIDE has as input the system's representation in Data

flow diagrams (DFDs). DFD has four types of elements: data flows, data stores, processes, and external entities. STRIDE systematically identifies possible threats in the interactions and also generates possible security recommendations.

According to Leveson and Thomas [3], since STAMP applies to any emergent property, STPA can be used for any system property, including cybersecurity. We use the extension of STPA with STRIDE to identify cybersecurity loss scenarios proposed by De Souza et al. [5]. We map the STPA control structure to the STRIDE DFD and then identify loss scenarios and recommendations due to cybersecurity threats.

C. Related work

In LedgerMail [6], CryptaMail [7] and Swiftmail [8] every email sent is a blockchain transaction. In this way, all transmitted information is stored immutably on all network nodes. In the context of national defense, the storage of emails is an undesirable feature, as the aim is to keep them on the network to a minimum time needed. Even if a message is encrypted if it is available on all nodes in the network, there is a possibility that a bruteforce attack discloses the content of the messages. Furthermore, none of the solutions are open source, SwiftMail uses its own blockchain and LedgerMail and CryptaMail use public blockchain. None of these features are desirable in a system within the scope of national defense.

In Invisible Ink [9], the solution also uses a public blockchain (bitcoin blockchain) to store email hashes, however, messages are stored in a separate database to provide the functionality of deleting messages to the user. Pretty Good Privacy (PGP) is used to verify the authenticity of a signature. The solution has two drawbacks. It uses a public blockchain. It employs an additional authentication mechanism to access the database, making the management of credentials more complex.

Some works use a distributed Public Key Infrastructure (PKI) based on blockchain, such as the ones proposed by Lewison and Corella [10], Axon and Goldsmith [11] and Yakubov et al. [12]. This type of design allows greater control over public key management by the user and motivated us to design our solution. However, they have not yet been applied specifically to a messaging system.

We are looking for a fully distributed messaging system, in which the user is able to delete messages and manage their public encryption keys. The related works found do not fully cover these requirements.

III. Development approach of messaging system

We propose a development approach for the messaging systems of a defense organization. The development approach uses blockchain to meet the security requirements of the messaging system and employs the extended STPA with STRIDE to identify requirements. The approach includes three activities:

- Use STPA and STRIDE to identify cybersecurity requirements for a system We perform the STPA analysis with the STPA extension that uses STRIDE, as proposed by De Souza et al. [5]. The output is the security requirements for the system's development.
- Design a system that can meet cybersecurity requirements using blockchain The designer analyzes the security requirements and uses blockchain and other technologies to create a system that aims to meet the requirements.
- Test the cybersecurity of the designed system Security tests are carried out on the system and an analysis is made of the types of requirements that were met due to the use of blockchain as part of the solution.

The approach has other development activities to build the system, such as implementation and testing, but they are regular activities found in the development process.

IV. Messaging system

The analyzed system is responsible for the secure sending of messages between military organizations distributed in the country. The activities for creating and testing a solution using blockchain are shown below:

A. Use STPA and STRIDE to list cybersecurity requirements for a system

The step has five steps. They are:

1) Step 1 – Define Purpose of the Analysis: In this step, we define the goal of the system, assumptions about the environment in which the system operates, the main stakeholders, the losses that must be avoided, the hazards, and the respective constraints.

The purpose of the system is to allow secure exchange of messages between users of an internal network of the armed forces, using email services with end-to-end encryption and public key infrastructure. The messages to be transmitted are confidential and serve as a support to decision-making by the armed forces.

To create the system, the following assumptions were considered:

- There is a server infrastructure distributed in military organizations (Intranet).
- There is a system for exchanging messages by email between military organizations, within the Intranet (Messaging System).
- Intranet mail servers can only send and receive messages to each other, they cannot contact other providers outside the Intranet.
- Each military organization has only one user.
- The messaging system is composed of: email servers and their registered users, PKI service, service for storing parameters about sent messages, and application for users (frontend).

By performing step 1 of STPA, we obtain the losses, hazards, constraints, and associations. They are described in Table I. All hazards are associated with at least one type of loss and have a constraint. For example, the hazard H1: "State that allows improper alteration of a user's messages or data (Tampering and lack of integrity)" is associated with loss L2:"Failure in the missions of the military due to security

problems in the organization messaging system" and has the constraint C1: "The system should not allow undue changes to users' messages".

TABLE I
LOSSES, HAZARDS, AND CONSTRAINTS.

Losses	Hazards	Constraints
L1: Loss of credibility of citizens in the Armed Forces due to an unacceptable number and severity of security issues that are brought to light	H1: State that allows improper alteration of a user's messages or data (Tampering and lack of integrity) [L2]	C1: The system should not allow undue changes to users' messages [H1]
L2: Failure in the missions of military organizations due to security problems in the messaging system	H2: State that allows access to the content of messages processed by someone other than the sender or recipient (Information Disclosure and lack of confidentiality) [L1, L2]	C2: The system must not allow messages to be read by people who are not the senders or recipients of the message [H2]
L3: Unacceptable number of users who are unable to send or receive messages	H3: State in which the system is not available to perform its activities (Denial of Service – availability and reliability) [L2]	C3: The system must always be available to carry out its activities (sending, receiving, and checking messages) [H3]
	H4: State that does not allow users to register in the system, send, receive or access the content of messages (to assure the mission) [L3]	C4: The system must always be working correctly and ensure that the user can operate correctly to send and receive messages [H4]

2) Step 2 – Model the Control Structure: The system is composed of six components. The control structure is shown in Figure 1. The responsibilities of each component are described below:

- User: End user of the system, responsible for sending and receiving end-to-end encrypted messages.
- Administrator: Responsible for keeping the system operational and for registering each military organization for the first time.
- Messaging application (MA): This is the component that interfaces with the user. In this component, it is possible to edit, encrypt, decrypt, sign, verify the signature, send, receive and check the history of messages. To encrypt a message or verify a signature, the integration component performs a query on the PKI and parameters storage component.
- Registration component (RC): Component used by the administrator to register users and provide application installation packages.
- Email component (EC): The email component is a set of common email servers on which users can send messages to their recipients. These servers are distributed in Military Organizations (MOs) to avoid single points of failure. It is possible to group MO according to needs and capabilities. Ideally, each MO should have its own email server installed in its infrastructure.
- PKI and parameters storage component (PPSC): The Public Key Infrastructure (PKI) and parameters storage component is responsible for storing public keys and messaging parameters (like hash, date, and time). MA uses PPSC data to validate the signatures of incoming messages and also to encrypt messages that will be sent.

The interactions necessary for the system to work are carried out through the control actions and feedback shown in Figure 1, for example, a user can use the MA to login, send or receive messages, among other activities. The downward arrows on the left between two components are the control actions while the upward arrows on the right are the feedback.

3) Step 3 – Identify Hazardous Control Actions: For each control action of the control structure, an analysis was performed to identify if it is hazardous or not in a given context. The result is a list of Hazardous Control Actions (HCA). An example of analysis is shown in Table II, the CA is "Forward message" and it is hazardous in the cases shown in the table, for example, when the system allows MA to send messages without encryption and signature.

TABLE II
CONTEXT TABLE FOR CA "FORWARD MESSAGE", FROM MA TO EC.

Message encrypted and signed	User authenticated	CA provided	CA not provided
Yes	Yes		The user is authenticated and the message is encrypted and signed. H4: The message must be forwarded for the system to function correctly.
Yes	No	The user is not authenticated and the message is encrypted and signed. H1: The user is not trusted, an attacker could be trying to use the MA to gain access to the User's computer.	
No	Yes	The user is authenticated and the message is not encrypted and signed. H2: Unencrypted messages can be read by network sniffers.	
No	No	The user is not authenticated and the message is not encrypted and signed. H1: The user is not trusted, an attacker could be trying to use the MA to gain access to the User's computer. H2: Unencrypted messages can be read by network sniffers.	

4) Step 4 – Identify Loss Scenarios: A loss scenario describes the causal factors that can lead to the unsafe control

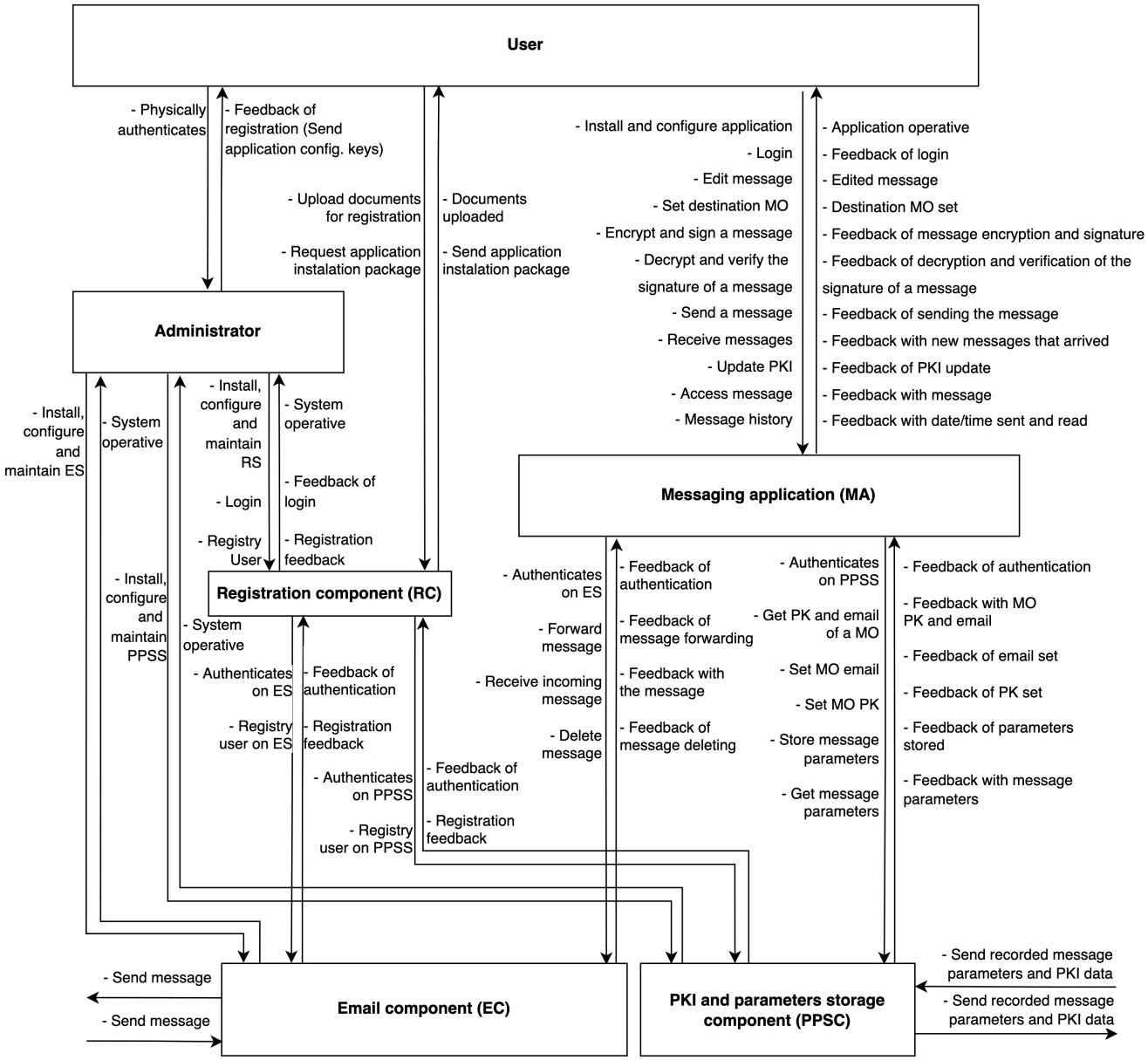

Fig. 1. Control Structure of the system.

action and to hazards. In the example of the CA "Forward message" when the "User authenticated" is *Yes* and "Message encrypted" is *No* a possible loss scenario is a misuse of the MA by the User, such as, a mistaking click. The requirement for the causal factor is that the MA must not allow sending messages without cryphering them. To evaluate security-related loss scenarios, the STRIDE extension to STPA was used, as in IV-A5.

5) Step 5 – Identify threats and vulnerabilities using STRIDE: The control structure was mapped to a DIEFD according to Table III, User and Administrator are external entities and the other elements are processes. Then STRIDE was applied to the DIEFD. all the links that make up the system were analyzed to identify possible threats that could generate loss scenarios. Requirements were created to address such threats. An example is shown in Table IV which represents the link between MA and PPSC. A possible STRIDE threat is "tampering". The generated requirement is that the data must be stored in a way that can not be modified.

B. Implement a proof of concept of the system to meet the cybersecurity requirements listed using blockchain

A proof of concept of the system was developed. We intend to verify if the proof of concept satisfies the requirements obtained in Section IV-A. A private Ethereum blockchain network was created, with *geth*, and a smart contract was deployed on the network. The smart contract is responsible for

TABLE III
MAPPING THE CONTROL STRUCTURE ELEMENTS TO DIEFD ELEMENTS.

Element	DFD category
User	External Entity
Adminstrator	External Entity
Registration component (RC)	Process
Messaging Application (MA)	Process
Email component (EC)	Process
PKI and Parameters Storage component (PPSC)	Process

storing the data of military organizations. The data includes name, email, public encryption key, and identification number. EC was simulated with free services from email providers. The decentralized application MA was created in Python. MA is able to communicate with the smart contract via the *web3* protocol and with the email service via *IMAP*. MA can receive and send messages with end-to-end encryption to other users on the registered network.

Some requirements that were met using blockchain are available in Table IV as protection against: spoofing, since transactions are signed with a strong encryption key; tampering, since not even administrators have the power to modify the blocks already added to the blockchain; denial of Service, since a distributed system is quite robust to this type of attack; and elevation of privilege, as no PPSC user account has more permission than another, there is no account that centralizes the greatest permission over the system.

C. Test the cybersecurity of the designed system

Security tests simulated known attacks. The attacks included (i) Man in the Middle (MITM) and sending forged packets, (ii) Distributed Denial of Service (DDoS), and (iii) Elevation of Privilege. Programs in Python and packet capture tool *Wireshark* were used.

The system proved to be robust to: (i) DDoS, as it is composed of Decentralized Applications (DApps), blockchain, and distributed email servers, (ii) MITM, because the messages are encrypted and signed, and (iii) Elevation of privilege in PPSC since the administrator can only create new users but cannot modify their data after the first password change by the user.

V. DISCUSSIONS

The users (MOs) have to register and operate. The registration takes place centrally by the administrator. The operation takes place in a decentralized manner by the users. Decentralized systems are more robust to DDoS attacks.

Assuming that most of the MOs are registered in the system's setup phase, the system is generally operating in a decentralized manner. After the setup phase, the administrator's work is eventual and not critical in case of a DDoS attack.

VI. CONCLUDING REMARKS AND FUTURE WORK

In this work, we proposed, created and tested a messaging system to be used by the Brazilian Ministry of Defense. We identified security requirements using the extension of STPA with STRIDE. The system uses blockchain to store the encryption keys used in PGP.

We develop the proof of concept to verify if it is possible to implement the proposed solution and check if the security requirements are met. The end user can edit, encrypt, decrypt, sign, verify signatures, update their PKI (smart contract) data and send emails.

We have shown that the solution can handle availability,

TABLE IV
STRIDE REPORT FOR LINK MA TO PPSC.

STRIDE Element	Description (Loss Scenario)	Recommendation (requirement)
Spoofing	PPSC may be spoofed by an attacker and this may lead to information disclosure by Messaging Application (MA).	PPSC must use a secure protocol to authenticate the Messaging Application, with encryption key exchanges.
Tampering	Data flowing across link may be tampered with by an attacker. This may lead to a denial of service, an elevation of privilege, an information disclosure by PPSC.	PPSC will store the message data (hash, timestamp, read and sent datetime, others) and this information is used to validate all incoming and outgoing messages. It is necessary that PPSC be developed in a way that prevents such information from being modified by anyone, even an administrator (who may have been hacked) or by the user himself (in an attempt to repudiate his message).
Repudiation	PPSC claims that it did not receive data (Set MO email, Set MO PK, Store message parameters, Get message parameters, Get PK and email of a MO, Authenticates on PPSC).	PPSC must log all transactions.
Information disclosure	Data flowing across the link may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations.	The data exchange protocol between the Messaging Application and PPSC must use encryption.
Denial of service	PPSC crashes, halts, stops or runs slowly; in all cases violating an availability metric. An external agent interrupts data flowing across a trust boundary in either direction.	PPSC must have link and asset redundancy as well as protection from DoS and DDoS attacks.
Elevation of privilege	An attacker may pass data into Messaging Application (MA) in order to change the flow of program execution within PPSC to the attacker's choosing. Messaging Application (MA) may be able to remotely execute code for PPSC.	The design of PPSC must be done in a way that avoids the possibility of elevation of privilege, such as strong password policies to administer, whitelist of MAC addresses or even that a possible successful elevation of privilege cannot have serious consequences for the system. Consider using distributed systems with transaction consensus mechanisms.

tampering, spoofing, and elevation of privilege threats.

A possible future work is to make the smart contract so that a subset of MOs is able to create other MOs. In this type of solution, there will be no need for the administrator.

The next steps are to carry out more security tests and implement other mechanisms that meet the requirements but were not implemented in the proof of concept, such as user authentication.

REFERÊNCIAS

[1] BISWAS, Baidyanath; GUPTA, Rohit. Analysis of barriers to implement blockchain in industry and service sectors. Computers & Industrial Engineering, v. 136, p. 225-241, 2019.

[2] ØLNES, Svein. Beyond bitcoin enabling smart government using blockchain technology. In: International conference on electronic government. Springer, Cham, 2016. p. 253-264.

[3] N. Leveson J. Thomas, "STPA Handbook," 2018.

[4] Microsoft. Uncover security design flaws using the STRIDE approach. MSDN Magazine; November 2006.

[5] De Souza, N. P., César, C. D. A. C., de Melo Bezerra, J., Hirata, C. M. Extending STPA with STRIDE to identify cybersecurity loss scenarios. Journal of Information Security and Applications, v. 55, p. 102620, 2020.

[6] LedgerMail (2018). Ledgermail home description. https://ledgermail.io. Accessed: 2022-07-22.

[7] CryptaMail (2014). http://www.cryptamail.com. Accessed: 2022-07-22.

[8] NewsBTC (2016). Using blockchain technology for email verification. https://www.newsbtc.com/news/john-mcafee-swiftmail-using-blockchain- technology-for-email-verification/. Accessed: 2022-07-22.

[9] Lazarovich, A. (2015). Invisible Ink: blockchain for data privacy. PhD thesis, Massachusetts Institute of Technology.

[10] Lewison, K. and Corella, F. (2016). Backing rich credentials with a blockchain pki. Pomcor.com.

[11] Axon, L. and Goldsmith, M. (2017). Pb-pki: A privacyaware blockchain-based pki. In SECRYPT, pages 311–318.

[12] Yakubov, A., Shbair, W., Wallbom, A., Sanda, D., et al. (2018). A blockchain-based pki management framework. In The First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block) colocated with IEEE/IFIP NOMS 2018, Tapei, Tawain 23-27 April 2018.