Systemization of Knowledge:
Trust as a Concept in Computer Security

name: Jeremi Stillmark
department name: Chair of IT Security
name of organization: *Technische Universität München*
city, country: Munich, Germany
email: jeremi.stillmark@tum.de

***Abstract*— Trust is crucial in our society to cooperate and rely on others. As nowadays a lot of time is spent online the concept of trust from a personal perspective with two individuals trusting each other, shifts to a more complex concept of trusting an online system and anonymous individuals. Although we trust different people and systems all the time the true definition of trust is hard to pin down. Which makes it even more difficult to work with trust in computer security where we need precision and clarity. Therefore, this paper systematizes and categorizes the term trust in computer security.**

In the first part the definition of trust is explained and common misconceptions are being cleared, by presenting multiple definitions of trust and differentiating the term from other concepts. After introducing important features of trust, the paper focuses on the applications and usage of trust in Computer Security. Those applications are explained by defining common threats that involve exploiting trust and categorizing trust architectures into centralized, decentralized and hybrid trust models. Additionally, the paper shows that the new "trustless" trust architecture, is not yet an alternative to other trust architectures.

Keywords—computer security, trust architecture, "trustless" trust

I. Introduction (*Definition of Trust*)

As trust surrounds us in every day-to-day activity it also plays a key role in Computer Security. From establishing and maintaining a trusted connection to the exploitation of trust through social engineering. When talking about the concept of trust in computer security it quickly gets very confusing how each author interprets trust. There are misconceptions and contradictory definitions which we will systematize in this paper. Additionally we will differentiate some terms from trust, that are often used interchangeably and show how trust is established.

To start we should have a look at the definition of trust by the Cambridge Dictionary which states that to trust someone is the same as "to believe that someone is good and honest and will not harm you, or that something is safe and reliable". [1]

In this definition we already see that there are different scenarios in which you can use the term trust. The most instinctive one is the scenario in which one person, also called trustor is confident that another person, called trustee will act benevolent. In the second scenario the trustee is switched for a system or object, which means that the trustor believes a system or object will act beneficially for them, while being save and reliable. Although this definition of trust is well suited for the general public, it is not complete when we apply it to Computer Science. Another definition of trust which defines trust as "the willingness of a party to be vulnerable to […] another party, based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other part" [18] is stated in the book "The Blockchain and the New Architecture of Trust" written by Kevin Werbach. In this definition the focus is shifted on three main aspects.

The first one is making yourself vulnerable to another entity, which is also the main reason why trust needs to be discussed in Computer Security. Because without a vulnerability or a threat, there would not be a need for security. The second point is the need of an action performed by the trustee, that benefits the trustor. This part is already included in the first definition that was presented and stays the same when looking at definitions of trust in other fields. [52] The third part of the definition describes the absence of the ability to monitor or control the trustee while it performs the expected action. It correlates with the aspect of making yourself vulnerable to the other party, because instead of verifying the truthfulness of the other party, you hope that they will perform the expected actions.

While describing these two definitions we can already see that the term trust is very close to similar concepts as reliance, confidence, or faith. To avoid any misinterpretations, we should first distinguish between those terms.

II. Similar Concepts

There are multiple ways to separate the concept of trust from other terms. One of them presented by Niklas Luhmann suggests using the term trust when describing interactions between (two) people and to use the term confidence when talking about interactions between people and systems. [2] This approach can be understood better if we look at the difference between both terms. Even though they are sometimes used interchangeably, there is a slight difference especially regarding the risk that a trustor has in a trustee. If someone has confidence in something the person is convinced that the thing will act in the way, it wants it to act. Machines usually act in the way we want them to act and even if they fail to meet our expectations it is never intentional, but rather by mistake. On the other hand, if someone trusts somebody, there still might be a high risk that the trustee fails to meet the expectations of a trustor intentionally, for example to sabotage or harm the trustor.

It is important to note here that in our modern era both definitions might fit at the same time. While using an app you have confidence in the system that none of your chats, pictures or other personal data will be leaked or handed over to third parties. At the same time, you trust the person that you are interacting with, to be the same person as their profile shows.

Another term we need to distinguish from trust is reputation. Jingwei Huang and David M Nicol have given a good distinction between both describing trust as a relationship between two entities and reputation as the subjective opinion of multiple users or a society on a person or system.[3]

The trust we have in someone might not always be built through good memories and positive interactions but also through the reputation that the person has built in a specific group.

Most of us would probably use an app that has a good reputation, based on user reviews or recommendations from friends. We will not use an application that hasn't got a good reputation, regarding both user and security features, although it doesn't necessarily mean that a higher reputation also leads to a more trustworthy app.

Here we can also distinguish between referral trust, which is the ability of Bob to recommend Alice an app and functional trust which is the trust in the functionality and security of the app. [11]

III. Trustworthiness

In the example above We stated that a system can be trustworthy, but until now we haven't defined what that means. Stating that an entity is trustworthy is the same as being willing to trust that entity. Usually, the trustworthiness of entities in a system is determined through their reputation, prior experiences and the confidence that we can have in the abilities of that entity. The trustworthiness of each entity needs to be determined before establishing a trusted connection between them. To determine if a system is trustworthy or not might be quite a challenge as trust is a psychological state and not a physical construct. Nevertheless, the trustworthiness of an application can be determined by using security assurance which measures the degree of confidence that you can have in a system. [6]

IV. Distrust/ Untrust/ Mistrust

If a trustor believes that a trustee is trustworthy, he or she might trust them. Sometimes this estimation might be false which then can lead to misplaced trust. It is important to mention that the trustee does not particularly has to have bad intentions and might just not meet their expectations. An app's security might have been not as good as estimated, which means that the system itself never had "bad intentions" it just didn't fulfill all requirements. [41]

One of the most popular data leaks for example was the latest Facebook data leak where data of over 500 million Facebook users was published in a hacker forum and was accessible for everyone.[8] The users trusted Facebook with their personal data, because they perceived it to be a trustworthy system, but the trust was misplaced. Just as every company, Facebook too needs the trust of their users to be able to function. Additionally as the trust of a lot of users might be lost, the cooperation threshold might increase as some people might not perceive the system as trustworthy anymore.

Distrust, mistrust and untrust are three terms that sound similar and sometimes are used interchangeably, nevertheless we should clarify their definition, before discussing the concept of trust in Computer Security.

Distrust defines a trustors disbelief in the trustees' intentions regarding him/her. This complete lack of trust in the trustee is usually traceable back to a prior negative experience that the trustor had with the trustee. If the trustee exploits the trust of the trustor once, it is very unlikely that the trustor would be willing to trust that person again. [42]

Another commonly mixed-up term is untrust which is defined as a very low trust level of the trustor for the trustee. Untrust is usually based on the lack of faith in the abilities or intentions of the trustee. As can be depicted from Figure 1 the trust level of untrusting someone is higher than distrusting someone.[5] Nevertheless, the cooperation threshold is not reached by neither of the expressions.

The term that exceeds the cooperation threshold is mistrust, which is defined as the lack of trust in someone due to instincts or feelings. In most cases the trustor trusts the trustee, even though they are not fully convinced or have a bad feeling about the situation. As shown in Figure 1, mistrust is very close to actual trust, which can be explained by the same outcome of both. For the outcome of a cooperation between two entities usually it is insignificant if one of them mistrusts the other one as long as both of them cooperate. It is important to note that the trust level can change over time, which can either lead to a cooperation or a breakup between both parties involved.

Too sum up those three terms it is best to look at the chart in Figure 1, where the trust level is shown for each of the terms. It is difficult to measure trust, because there are no units of trust, and it mostly depends on each subjective view, how high their cooperation threshold is. For one person the cooperation threshold might be a lot lower than for another one. And even for the same person the cooperation threshold might change depending on the partner that they are interacting with. [9]

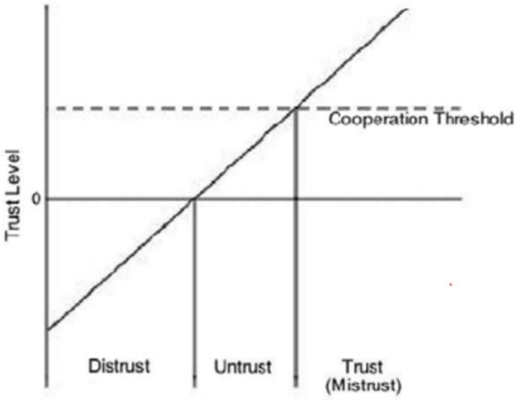

Figure 1: Trust levels of Distrust, Untrust, Mistrust [7]

V. Transitivity

We already scratched the surface of the different factors that might influence or persuade someone to trust somebody, one of which was the recommendation by friends or online reviews. This attribute of trust is called transitivity. If person B trusts person A and additionally person A trusts person B and C, because of the transitivity of trust person B also might trusts person C. The Robin Sage Experiment conducted by Thomas Ryan from December 2008 till January 2009 is a perfect example of how the transitivity of trust works in social networks and how it can be exploited. The main goal was to get access to certain information that should not be accessible, by persuading people. The experiment worked in the following way: The author created fake accounts on several social media platforms namely LinkedIn, Facebook, and Twitter under the name of Robin Sage an imaginary woman in the age of 25. Additionally, the profiles had pictures of an attractive young female to show the importance that appearance and sex have when determining if someone is trustworthy or not. The author befriended Robin with important people like a lecturer at the NASA Ames Research Center, who later sent her an unreleased paper he was working on, to get her opinion on it. If Thomas Ryan had bad intentions, he could have easily published the paper under his name and get all the credit, even though he never researched anything in this field. To increase the trustworthiness of the profile, the author added certain attributes like a graduation at the MIT and 10 years of work experience in the Cyber Security field. This led to her receiving several job-offers from companies like Lockhead Martin, one of the biggest arms companies in the world, and several others. The experiment is a perfect example for how the transitivity of trust can be exploited, because the whole networking process of Robin was based on it. It was enough that person A had added her as a friend and another person B, which is friends with person A could see that they have a mutual friend which increased the trustworthiness of Robin. The transitivity of trust as shown in the example forms triangles of trusted people. One person wrote the following message to Robin Sage: “I've never met you, but I saw you had Marty on your Facebook list, so that was good enough for me.” [22] Which shows that the “victims” had clearly fallen for the trap and trusted the fake profile because of the transitivity of trust. It is important to mention that even though the transitivity could be exploited in this example, it is not a prove that this kind of trust triangle is always possible. As the outcome mostly depends on the circumstances and the interacting parties. [12]

VI. Uncertainty

One of the most important issues when talking about trust is uncertainty, as trust can only exists in environments with some level of uncertainty. If every entity of a system knows everything, meaning there is no uncertainty about the future and therefore every action of other entities is predictable, trust wouldn’t exist. Therefore, trust is just a “filler” for the information that we do not have so that we can do predictions about the future as precise as possible and cooperate with others without having access to every information.

The proverb “Trust but verify” in this case would be senseless as you can either trust someone and accept the risk of getting betrayed or verify the information and do not trust. Trusting and verifying information at the same time does not work, because by verifying the information you eliminate the need for trust.

A good example to present a situation in which trust is crucial is the commonly known Prisoner’s Dilemma derived from game theory.

The prisoner's dilemma is a classic example of a situation in which two individuals must make a decision that will affect both of them, but in which neither has complete information about the other's state of mind. As described previously the trust can here be used as a “filler” for the missing information. The scenario is often framed as follows: two individuals have been arrested for two crimes and are being held in separate cells. The police do not have evidence to convict them of the main crime, but they do have enough evidence to convict them of a lesser crime. The detectives offer both the same deal: if one confesses and the other remains silent, the one who confesses will walk away as a free man, while the other will receive a harsher sentence. If both confess, they will both receive a moderately harsh sentence. And if both remain silent, they will both receive a lesser sentence. Each prisoner must choose whether to confess or remain silent, but neither knows what the other will do. This creates a situation where both individuals would be better off if they both remain silent, but if one assumes the other will confess, it becomes rational to confess first. The prisoner's dilemma can be applied to many real-world situations where two parties must make a decision that affects both, but where neither party has complete information about the other's decision. [13]

Here both suspects have very little information about the actions of the other suspect which leaves them with the choice to either trust the other suspect or to act in best self-interest.

It is important to note that just like in the prisoner’s dilemma in other scenarios “trust [also] implies a decision. Trust can be seen as a process of practical reasoning that leads to the decision to interact with somebody.” [14]

VII. The difficulty when working with Trust

Now that we have established a fundamental structure of the meaning of trust and cleared some common misconceptions, we can have a closer look at the usage of trust in computer security. As mentioned in the previous section we only need trust in environments where we have a lack of information. A similar concept applies for security, because you do not need any security if you can trust every entity in a system that they will act according to certain rules and restrictions. Nevertheless, there is always a need for security and trust, as long as you are not able to trust every entity fully and do not have access to all the information that you need.[15]

Working with trust in computer security is very difficult, mostly because security needs precision and clarity so that the system can guarantee certain services, but as trust is a psychological construct with lots of different definitions as described in the chapter before it is difficult to provide precision and clarity in trust.[16] And even if the definition of trust is clear, it might change for a specific context. When we define trusted code for example, it might mean that the code is trusted and therefore runs with high privileges. In other words, code that is not trusted runs with low privileges. But at the same time trusted code might also be understood as code that is trustworthy and therefore runs with higher privileges. This is a common misconception as the code does not necessarily needs to be more trustworthy only because it runs with higher privileges then another code. [17] In the following chapters we will not focus on trust in the code, but rather how trust is modelled in networks and how it effects the security of those networks.

VIII. Threats

One of the most known attacks where an attacker can listen to a connection between two users is the man-in-the-middle attack. We can describe this type of attack best with an example using Bob and Alice two friends that want to use a chatroom they found which will deliver each other's messages securely through the internet. Bob trusts Alice and Alice trusts Bob that's why they establish a connection and chat. But without their knowledge this connection is being hijacked and an attacker (in this example Malory) listens to their chat. If later Bob or Alice find out about it, they will most probably lose confidence in the system and stop using this chatroom to interact.[39] That's why as the developer of a networking system, it is important to not only establish a secure connection but also maintain it and not allow such attacks to take place so that more people will trust this service and use it.

A phishing attack describes an attack where malicious code, or a link is sent to a target. The message is usually written in a way that gains the targets trust, by for example impersonating as another employee or client. Once the target clicks on the link the attacker can get crucial credentials that can lead to him or her gaining access to financial information of the victim. [40]

When talking about threats that involve impersonating someone else, we must mention social engineering, which is a tactic that is used by attackers to manipulate someone into giving them crucial information or access to resources that they are not allowed to access. The attacker, usually disguised as a trustworthy person tries to gain the victims trust and then uses the victim to get what he or she wants.

These threats show how important it is to establish a trusted relationship over the internet without having someone listening in on the messages and at the same time being able to authenticate both parties as trustworthy.

IX. Trust establishment

To establish a trusted connection between two entities there needs to be a matching between already trusted instances and the new entity. A good example might be the verification with an identity card, which in the digital world might be the authentication of a user through a trusted third-party. Additionally, to the verification of the identity, the data exchanged between two entities needs to be protected from unauthorized access and manipulation.[19]

Trust establishment in computer security is a multi-layered process that involves creating and maintaining a secure connection between two parties at various levels of communication.

At the transport layer, the most common protocols used to establish trust are Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). These protocols use a combination of asymmetric and symmetric encryption to secure the communication between two entities.

Asymmetric encryption, also known as public-key encryption, uses a pair of keys, a public and a private key, to encrypt and decrypt the data. The public key is used to encrypt the data, and the private key is used to decrypt it. The main advantage of asymmetric encryption is that it allows for secure communication without the need for the parties involved to share a secret key in advance. [33]

Symmetric encryption, on the other hand, uses the same key for both encryption and decryption. This key must be securely exchanged between the parties involved before the communication can begin. Symmetric encryption is generally faster and more efficient than asymmetric encryption, making it suitable for encrypting large amounts of data. [34]

TLS uses asymmetric encryption to establish a secure communication channel and then use symmetric encryption to transfer the data. The process starts by the client and server negotiating to agree on the version of TLS protocol they want to use, the cipher suites and the set of security parameters they will use. Then, the client and the server use their asymmetric key-pairs to authenticate themselves to each other and establish a shared secret key (session key). This is done through a process called key exchange, typically using algorithms like RSA, DH, ECDH.

Once the shared secret key has been established, both entities use it to encrypt and decrypt the data using symmetric encryption. This ensures that the data cannot be read or modified by an attacker who intercepts the communication. And by using a new session key for each session, the protocol ensures that even if one session key is compromised, the attacker will not be able to decrypt any previous or future sessions.

At the network layer, trust can be established using Virtual Private Network (VPN) protocols such as Internet Key Exchange (IKE) and the Internet Security Association and Key Management Protocol (ISAKMP). These protocols are used to establish a secure tunnel between both entities, allowing them to exchange data securely over a public network.

At the application layer, trust is established using various authentication and authorization methods, such as single sign-on (SSO) and multi-factor authentication (MFA). These methods are used to verify the identity of the parties involved and to ensure that only authorized individuals have access to sensitive data.

As the name already states the multi-factor authentication works by authenticating the user through multiple ways. The most common use case is the login process at an online banking system, where you need to login with your correct credentials and password. Additionally, you must open an app on your phone, where a message needs to be accepted, which states that you are the person that tries to login to the account.

It is necessary to note that depending on the circumstances and the trust level that is shared between all parties, the trust establishment process can vary. If you create an application to be able to chat to your family you might use lower security measures, then a banking system would use when authenticating users.

A crucial part of trust establishment via the internet is the use of a Public Key Infrastructure (PKI). PKI is a system that uses digital certificates and a certificate authority (CA) to authenticate the identities of the parties involved in a communication. A CA is a trusted third party that issues digital certificates to clients and servers, which they can use to prove their identity. PKI creates a secure environment for communication by providing authentication and encryption, which are essentials for the establishment of trust. [35]

X. Zero trust

One commonly used trust model when designing a secure network is the zero-trust model. "Never trust, always verify" is a common proverb that describes how the zero-trust model can be understood. [20] As the name already suggests the network does not trust any user, device, or system until it is verified that the user can be trusted. If a user wants to gain access to certain resources, the system checks the device's identity, role, and current level of trust, before granting access. This means that all users, devices, and systems must be continuously verified and authenticated before being granted access to network resources.

There are several key principles that the zero-trust model employs to achieve this. One of them is the multi-factor authentication which was already mentioned in the section above.

Another principle is the least privilege principle which ensures that users only have a minimal level of access required to perform their job functions. This reduces the risk of accidental data breaches or malicious actions. The goal here is to limit the damage that can be done when a hacker gains access to the user's device or account. [43]

An important principle in the zero-trust model is micro-segmentation, which is the process of dividing a network into smaller, isolated segments. This makes it more difficult for an attacker to move laterally within a network and reduces the risk of a data breach by isolating sensitive data from less-sensitive data. Micro-segmentation is typically accomplished using firewalls and other network security devices and allows administrators to create different security policies for different parts of the network. [44]

As stated before the zero-trust model also must ensure that the users, devices and systems in a network are continuously verified. This is accomplished through continuous monitoring and analysis by monitoring the activity on a network to detect and respond to security threats in real-time. This includes monitoring for unusual or suspicious activity and taking action to stop or mitigate a threat if it is detected.[31]

XI. Trust architectures

Before presenting some of the most important trust architectures used nowadays, we need to clarify the importance of those models. As described earlier trust is essential in nearly every day-to-day activity. If we now want to have a look at more than one relationship, we might quickly lose an overview due to the number of different trust relationships between each entity. That's why there are some trust architectures which specify how the trust is distributed between each entity. Without trust architectures, it would be difficult to establish secure communications and maintain confidentiality, integrity, and availability of data and systems. Additionally trust architectures provide a broader view on the network's trust distribution, which makes it easier to understand how the network works and develop it further according to specific requirements. A robust trust architecture can help prevent fraud, protect against cyber-attacks, and ensure compliance with regulations. [36]

To get a better understanding of each trust architecture we categorized them into three different groups, namely: centralized, decentralized and hybrid trust models. [21]

The centralized trust models always have an instance that is between both entities that want to establish a trusted relationship. This instance might be a third-party that establishes the connection between both entities and ensures that the connection is secure. This intermediary entity usually is also responsible for the authentication of both identities, so that a trusted connection can be established, although both entities never trusted each other. As can be seen in Figure 2 the intermediary third-party is connected to each entity and as there is no direct trust between the entities, there are no direct connections between the individual entities. It is important to add that for such a system to work both entities must trust the third-party. Centralized trust models are often used in systems where there is a need for a high degree of control over trust relationships, such as in financial systems. [30]

Decentralized trust models, on the other hand, rely on distributed trust management. This approach uses multiple entities, called trust anchors, that work together to establish trust. Trust anchors can be individuals, organizations, or devices that are trusted by the parties in the system. Each entity needs to trust the other entities and the system to function. In Figure 2 the example of a peer-to-peer trust architecture is presented, which presents a possible way to model these trust models. As there is no third-party that controls the network it is not as easy to make changes as it is the case in centralized trust models. Decentralized trust models are often used in systems where there is a need for a high degree of scalability, such as in peer-to-peer networks.[29]

The third trust model type is the hybrid trust model which is a combination of both centralized and decentralized trust models. This provides a balance between the high scalability of decentralized models and the controllability of centralized models. As can be depicted in Figure 2 by the example of the leviathan trust architecture, the third party provides a frame and does not act as an intermediary like in the centralized trust models. The selection of the appropriate trust model depends on the requirements and constraints of the system and the trade-offs that are acceptable. It is important to note that none of the mentioned trust model types is always better than the others. [37]

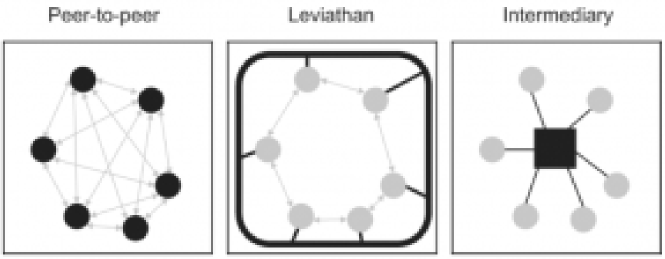

Figure 2: Peer-to-peer, Leviathan and Intermediary trust architectures in comparison [10]

XII. Peer-to-peer trust

The simplest trust architecture is direct or peer-to-peer (P2P) trust which is the trust between two or more entities just as described earlier, Alice trusts Bob and Bob trusts Alice. This trust architecture mostly relies on mutual trust between entities. As it is difficult to trust someone without an authentication of that person this model is very limited especially for the online usage. The peer-to-peer trust architecture is a decentralized trust model because there is no intermediary party that controls the system.

P2P trust architecture relies on the use of reputation systems to establish trust between peers. Reputation systems are used to evaluate the trustworthiness of a peer based on its past behavior in the network. For example, a peer that has a high reputation score is more likely to be trustworthy than a peer with a low reputation score. [27]

To establish trust, peers exchange information about their reputation scores with each other. The reputation scores can be used to identify trustworthy peers and to avoid interacting with untrustworthy peers. Reputation scores can be based on a variety of factors, such as the number of successful interactions a peer has had with other peers, the number of resources (e.g., bandwidth, storage) that a peer has contributed to the network, or the length of time that a peer has been active in the network. [23]

P2P trust architecture also relies on the use of digital signature and encryption to secure the communication between peers and make sure the authenticity of the messages exchanged. It is well suited for systems where a high degree of scalability is required, such as in file sharing, decentralized social networks, and distributed computing applications.

XIII. Third-party trust

A centralized trust architecture uses a third-party also called intermediary, which both entities trust, without having to trust each other. An example could be an online purchase where you pay via PayPal and do not trust the vendor and he or she doesn't trust you, but both of you trust that the PayPal service will deliver the money from one account to the other. [26] The trusted third-party, in our example PayPal, not only builds a bridge between the entities, but also provides additional services such as security, authentication, and dispute resolution.

The importance of this trust architecture becomes obvious when you look at the way that a trusted connection is established at every Google search or page that is visited on the internet. The concept that establishes secure communications and trust relationships over an insecure network such as the Internet is called Public Key Infrastructure (PKI).

Because of the PKI the user only needs to trust the browser and the third-party, in this case a certificate authority (CA), which creates a digital certificate. A digital certificate is an electronic document that contains a public key for the certificate holder and is used in combination with a private key for the certificate holder for secure communication. The certificate holder can use the private key for decrypting the message and the other party can use the certificate's public key for encrypting the message.

The certificate authority acts as a trusted third party, which establishes trust between the parties by vouching for the identity of the certificate holder. To understand the PKI there are two trust models which need to be mentioned. Both are centralized trust models that are build on a third-party trust architecture.

The first one is the hierarchical trust model, which describes a trust model in which a certificate authority (CA) that signs its own digital certificate by stating itself as the instance who should be trusted and who can verify the trust. [24] In a bigger system this would mean that every public key is verified by the same CA. The problem with this trust model is that if once someone can fake the certificate all certificates that were "signed" by this CA will be invalid.

Therefore, there is the distributed trust model which can surpass this limitation, by setting one root CA which then has multiple smaller CAs with intermediate certificates that sign each end-entities digital contract. As can be seen in Figure 3 the root certificate is trusted by the intermediate certificates which are then again trusted by the user.

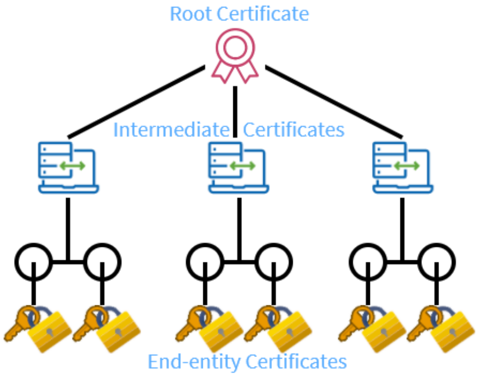

Figure 3: The trust architecture of certificate verification in the Public Key Infrastructure [25]

XIV. Leviathan trust

The leviathan trust architecture is a theoretical model that was inspired by the concept of the Leviathan from the Bible, a sea monster that can control the chaotic and unpredictable sea.

It is a hybrid trust model, as it still uses a third-party (authority) to control the network, but the third-party is not an intermediary instance like in the centralized trust models. Additionally, the entities communicate directly and therefore also establish trust directly. Just as in the analogy of the leviathan from the bible, the third-party is able to control the chaotic and unpredictable actions of the entities in the network. Because the trusted third-party operates from the background it ensures that each entity holds up to their promises.

The authority is allowed to punish the entities that do not hold up to their promise and therefore become untrustworthy. The leviathan trust architecture can be compared to a state that can use law enforcement or the military to make sure that everyone follows the rules. The trust between two entities is then a lot higher even if they never interacted with each other, because both parties are confident that if the other party misuses the trust, they can get a penalty for that. This of course reduces the risk that must be taken when trusting a stranger and makes it easier to cooperate. [38]

The leviathan trust architecture is not widely spread in real-world applications, as it is only a theoretical model. Nevertheless, parts of the concept can be observed in online social networks, like Facebook or TikTok where algorithms and policies ensure that a trusted communication is possible for every user. Additionally, entities that do not follow those policies are removed from the system.

XV. "TRUSTLESS" TRUST ARCHITECTURE

One of the newest trust architectures is the "trustless" trust architecture, which is a decentralized trust model, as there is no trusted third-party that controls or verifies each user.

From the name itself it might be wrongfully concluded that this architecture type replaces trust with something else or works fully without trust. But in fact, the trust is only shifted from trusting another person or third-party, to trusting the code and the system.

One of the most common applications of that trust architecture type is the blockchain technology, which is a distributed ledger that is maintained by a network of computers, rather than a single central authority.

As the name suggests you can imagine the Blockchain as a chain of blocks. Each block on the chain contains a record of multiple transactions. It is important to note that it is possible to add new blocks to an existing chain. Once a block is added to the chain, the information it contains is set in stone and cannot be altered. Additionally, each block contains a record of all previous transactions on the chain, what makes it increasingly difficult for anyone to alter or tamper with the information on the blockchain.[4]

When a transaction is made on a blockchain, it is broadcasted to the entire network, where it is verified by multiple computers, also called nodes. These nodes use mathematical algorithms to confirm, that the transaction is valid, and that the sender has the necessary funds. Once a transaction is verified, it is added to the next block, which is then added to the chain.

This process of verification and addition to the chain is called "mining". Miners are incentivized to provide their computing power through the issuance of new cryptocurrency for creating new blocks. A popular example for such cryptocurrencies might be Bitcoin or Ethereum.

As the blockchain has a decentralized nature, there is no single point of failure, making it more resistant to hacking and other forms of tampering. [28]

With that in mind you might think that the Blockchain presents a new type of trust that is superior compared to the others, because you do not have to trust any entity anymore. But unfortunately, all parties still need to have confidence in the code and the cryptography that replaces the trust.

Additionally, it is important to note that there is a vulnerability namely the 50 plus one attack which might be accomplished with the use of quantum computers in the future. It works as follows, as the ledger of the blockchain is public and a copy is to be found on every device in the network, those copies on each device need to be updated. When the update is done the device trusts that the "true" version of the ledger is always the longest one, meaning with the highest number of blocks on it. Because of the high computational power of quantum computers, they might in the future be able to create a longer ledger then the "true" one and therefore fool all entities into trusting their ledger.

Kevin Werbach also shows an already existing vulnerability in his book "The Blockchain and the new architecture of trust", where he shows that the trust architecture still is not flawless and shows a failed "experiment" from 2016. When a German start-up created a new crowdfunding system which was a decentralized autonomous organization (DAO). The DAO should distribute the funds by "democratically" letting the users vote for certain projects. Sadly, the whole system failed when a hacker found a bug in DAOs code and stole more than one third of the funds. As there was no authority which could make decisions and take immediate action, chaos broke out between the users. Fortunately, the funds could later be regained by an intervention of the creators of Ethereum which had to take control. [32]

We have shown that the new trust architecture has an important advantage in comparison to the common trust models, by replacing the trust with code and cryptography. Nevertheless, an implementation of such a trust type is quite difficult in comparison to the other models, what can be seen by the previously provided example.

XVI. DISCUSSION

In the following we will discuss what advantages and disadvantages each of the trust types, that were described earlier have and take a closer look at the applicability of the "trustless" trust architecture.

The decentralized trust models are applicable for networks where each entity trusts one another and can make sure that no malicious entity is able to join the network. The biggest advantage of those trust models, at the same time is their biggest disadvantage. As these trust models do not use a third-party it makes the network cheaper to maintain, but at the same time creates a lack of security and control over the network. Additionally, these kinds of trust models are easily scalable as long as each entity can be trusted.

Another trust model is the hierarchical trust model which requires a trusted third-party, that creates and signs the certificates for all entities. The biggest disadvantage of this model is the cost of creating and maintaining such an infrastructure which was previously described with the example of the PKI. Additionally, every entity needs to sign up at the third-party which requires a new process. But the hierarchical trust model has a lot of advantages especially concerning the security and reliability of the communication between users. Because of the third-party involved, there is someone that is accountable for failures and for the deletion of entities that misused their trust.

The leviathan trust model as a hybrid trust model, combines the advantages and disadvantages of the decentralized trust models and the centralized trust models. It makes the network not as vulnerable to attacks as the decentralized trust models and provides easier scalability and direct connections between entities in contrast to the centralized trust models.

The newest trust model, "trustless" trust is still heatedly discussed especially when talking about the future applications of blockchain. Here no third-party is needed which saves a lot of costs in comparison to the centralized or leviathan trust models. At the same time the trust model is applicable for multiple users and could be used via the internet not like the distributed trust model. And it offers a very high level of security because the certificates that are created are hashed and therefore hard to decode without the right key. The "trustless" trust architecture still has some disadvantages though, the biggest one is the human aspect. For example, if you lose your key, all the data stored will be gone or not accessible anymore. Therefore, the trust model is not a model that a lot of people would trust. Additionally, to access any of the data or even to create a certificate that enables trust you need special systems like a wallet that are created by third parties. Which in conclusion means that you still need a trusted third-party, the only thing that changes, is the workload and tasks assigned to the third-party. But not only the human aspect is a big disadvantage of the "trustless" trust architecture. [45] Another vulnerability might come up due to the development in the field of quantum computing. As these entities have a much higher computational power then commonly used devices there might be the risk of the 50 plus one attack which would make it impossible to detect if the new updated copy of the ledger is correct or not. As long as no countermeasures against such attacks are to be found, the new trust architecture is not applicable and cannot be used.

Finally, we should add that there is no superior or inferior trust model in the sense that one will always be safer or more efficient. The choice which model to use is usually dependent on the situation, on the size of the system and on other factors like security risks or efficiency.

XVII. Conclusion

The purpose of this work has been to systematize and categorize the knowledge about trust in computer security. We accomplished it by firstly defining what trust means and distinguished it from similar concepts like confidence, assurance, trustworthiness and reputation. Additionally, we defined some important terms like distrust, mistrust and untrust and compared each trust level to another. With the help of some examples, we were able to show different attributes of trust, especially transitivity and uncertainty. With this foundation we had a look at the involvement of trust in computer security and stated the difficulty of working with trust in this environment. We had a look at how trust is established over the internet and what difficulties arise from that. After showing some common threats in computer security and scratching the surface of the zero-trust concept, we presented different trust models and discussed their advantages as well as their disadvantages.

The four trust models that this paper focused on where the centralized trust models, the decentralized trust model, the hybrid trust model, and the "trustless" trust model. We presented each model by providing examples and real-life applications like the Blockchain for the "trustless" trust model. After discussing the applicability of each of those trust models we concluded that none of them is "better" than the others as every trust model has their own context and applications in which they outperform the other models. We showed that the hierarchical trust model is commonly used nowadays to establish and maintain a trusted relationship over the internet and discussed if the "trustless" trust architecture might replace it in the future. Given the various problems that this trust model still has, it is not yet applicable for the daily usage and is not capable of replacing the other trust models without further changes.

The growing literature around trust and trust models in computer security indicates that a change of the way that trust is established online nowadays might take place. But there are still too many different definitions of the term trust even when we narrow it down to the usage of trust in computer security. As long as the term itself is not generally defined it is difficult to work with this simple yet so often misunderstood concept.

References

[1] https://dictionary.cambridge.org/de/worterbuch/englisch/trust

[2] Luhmann, Niklas. "Familiarity, confidence, trust: Problems and alternatives." Trust: Making and breaking cooperative relations 6.1 (2000): 94-107.

[3] Huang, J., Nicol, D.M. Trust mechanisms for cloud computing. J Cloud Comp 2, 9 (2013) https://doi.org/10.1186/2192-113X-2-9

[4] Deshpande,Advait,et.al."DistributedLedger Technologies/Blockchain: Challenges, opportunities and the prospects for standards." Overview report The British Standards Institution (BSI)

[5] Marsh, Stephen & Dibben, Mark. (2005). The Role of Trust in Information Science and Technology. ARIST. 37. 465-498. 10.1002/aris.1440370111.

[6] Cheshire, Coye. "Online trust, trustworthiness, or assurance?." Daedalus 140.4 (2011): 49-58.

[7] Marsh, Stephen & Dibben, Mark. (2005). The Role of Trust in Information Science and Technology. ARIST. 37. 465-498. 10.1002/aris.1440370111. (p.21)

[8] https://www.businessinsider.com/stolen-data-of-533-million-facebook-users-leaked-online-2021-4

[9] Jøsang, A. (2007). Trust and Reputation Systems. In: Aldini, A., Gorrieri, R. (eds) Foundations of Security Analysis and Design IV. FOSAD FOSAD 2007 2006. Lecture Notes in Computer Science, vol 4677.Springer,Berlin,Heidelberg.

[10] Werbach, Kevin. The blockchain and the new architecture of trust. Mit Press, 2018. (page 29 figure 1.2)

[11] Jøsang, A. (2007). Trust and Reputation Systems. In: Aldini, A., Gorrieri, R. (eds) Foundations of Security Analysis and Design IV. FOSAD FOSAD 2007 2006. Lecture Notes in Computer Science, vol 4677.Springer,Berlin,Heidelberg.

[12] Jøsang, A. (2007). Trust and Reputation Systems. In: Aldini, A., Gorrieri, R. (eds) Foundations of Security Analysis and Design IV. FOSAD FOSAD 2007 2006. Lecture Notes in Computer Science, vol 4677. Springer, Berlin, Heidelberg. (page 212)

[13] Erriquez, Elisabetta. Computational models of trust. Diss. University of Liverpool, 2012. (page 6)

[14] Pinyol, I., Sabater-Mir, J. Computational trust and reputation models for open multi-agent systems: a review. (page 10)

[15] Schneier B. Liars and Outliers : Enabling the Trust That Society Needs to Thrive. Indianapolis IN: Wiley; 2012.

[16] Gollmann, Dieter. "Why trust is bad for security." Electronic notes in theoretical computer science 157.3 (2006): 3-9. (page 8)

[17] Gollmann, Dieter. "Why trust is bad for security." Electronic notes in theoretical computer science 157.3 (2006): 3-9. (page 7)

[18] Ernst Fehr, On the Economics and Biology of Trust, Journal of the European Economic Association, Volume 7, Issue 2-3, 1 May 2009

[19] Noordergraaf, Alex. Enterprise Security: Solaris Operating Environment. Sun Microsystems Press, 2002.

[20] Buck, Christoph, et al. "Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust."Computers&Security110(2021):102436.

[21] Werbach, Kevin. The blockchain and the new architecture of trust. Mit Press, 2018.

[22] Ryan, Thomas, and Gabriella Mauch. "Getting in bed with Robin Sage." Black Hat Conference. 2010.

[23] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003), Linköping, Sweden, 2003, pp. 150-157

[24] R. Perlman, "An overview of PKI trust models," in IEEE Network, vol. 13, no. 6, pp. 38-43, Nov.-Dec. 1999, doi: 10.1109/65.806987.

[25] https://www.keyfactor.com/wp-content/uploads/Certificate-Chain-of-Trust-Sep-02-2020-08-15-34-48-PM.png

[26] Werbach, Kevin. The blockchain and the new architecture of trust. Mit Press, 2018.

[27] Aberer, Karl, and Zoran Despotovic. "Managing trust in a peer-2-peer information system." Proceedings of the tenth international conference on Information and knowledge management. 2001.

[28] Werbach, Kevin. The blockchain and the new architecture of trust. Mit Press, 2018.

[29] Gutscher, A. (2007). A Trust Model for an Open, Decentralized Reputation System. In: Etalle, S., Marsh, S. (eds) Trust Management. IFIPTM 2007. IFIP International Federation for Information Processing, vol 238. Springer, Boston, MA.

[30] S. Rizvi, J. Ryoo, Y. Liu, D. Zazworsky and A. Cappeta, "A centralized trust model approach for cloud computing," 2014 23rd Wireless and Optical Communication Conference (WOCC), Newark, NJ, USA, 2014, pp. 1-6

[31] T. Dimitrakos et al., "Trust Aware Continuous Authorization for Zero Trust in Consumer Internet of Things," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 2020, pp. 1801-1812

[32] Werbach, Kevin. The blockchain and the new architecture of trust. Mit Press, 2018.

[33] Schneier, Bruce. Applied cryptography: protocols, algorithms, and source code in C. john wiley & sons, 2007.

[34] Ferguson, Niels, Bruce Schneier, and Tadayoshi Kohno. Cryptography engineering: design principles and practical applications. John Wiley & Sons, 2011.

[35] "PKI: Implementing & Managing E-Security" by Steve Lloyd and Paul C. van Oorschot

[36] M. Patel, S. Bhattacharyya and A. Alfageeh, "Formal Trust Architecture for Assuring Trusted Interactions in the Internet of Things," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA,2019, pp. 0033-0039

[37] Werbach, Kevin. The blockchain and the new architecture of trust. Mit Press, 2018. (p.25-27)

[38] Werbach, Kevin. The blockchain and the new architecture of trust. Mit Press, 2018. (p.27)

[39] F. Callegati, W. Cerroni and M. Ramilli, "Man-in-the-Middle Attack to the HTTPS Protocol," in IEEE Security & Privacy, vol. 7, no. 1, pp. 78-81, Jan.-Feb. 2009

[40] Ramzan, Z. (2010). Phishing Attacks and Countermeasures. In: Stavroulakis, P., Stamp, M. (eds) Handbook of Information and Communication Security. Springer, Berlin, Heidelberg.

[41] Marsh, S., Dibben, M.R. (2005). Trust, Untrust, Distrust and Mistrust – An Exploration of the Dark(er) Side. In: Herrmann, P., Issarny, V., Shiu, S. (eds) Trust Management. iTrust 2005. Lecture Notes in Computer Science, vol 3477. Springer, Berlin, Heidelberg.

[42] Primiero, Giuseppe. "A calculus for distrust and mistrust." Trust Management X: 10th IFIP WG 11.11 International Conference, IFIPTM 2016, Darmstadt, Germany, July 18-22, 2016, Proceedings 10. Springer International Publishing, 2016.

[43] Haber, M.J. (2020). Zero Trust. In: Privileged Attack Vectors. Apress, Berkeley, CA.

[44] N. Sheikh, M. Pawar and V. Lawrence, "Zero trust using Network Micro Segmentation," IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Vancouver, BC, Canada, 2021, pp. 1-6

[45] Caradonna, Toni. "Blockchain and society." Informatik Spektrum 43.1 (2020): 40-52.

END