Conference Session: A11
Paper 8069

ZERO-KNOWLEDGE ENCRYPTION IN THE CLOUD: A SOLUTION FOR THE REMOTE FILE STORAGE

Ryan Luchs, ral95@pitt.edu, Mena, 1:00, Luke Sneeringer, lus31@pitt.edu , Sanchez, 3:00

Abstract—*With the advent of cloud computing, there has come a large increase in the outsourcing of data storage to remote servers, especially in business. Cloud computing is an advantage to businesses, eliminating the costs of constructing and maintaining their own computer infrastructures while providing a platform for users to create large quantities of mobily accessible, shareable data. Cloud computing also establishes a platform for sustainable, scalable computing for the future. Through the use of cloud computing, many resources can be conserved due to its centralized nature. Despite these assets, cloud computing poses a risk of data leakage, as it is difficult to verify the confidentiality and integrity of the cloud services of any given provider. As such, the ability for data to be accessed from the server side presents the possibility for any entity with access to the server to access and leak data. Zero-knowledge data encryption provides a solution in this scenario.*

A zero-knowledge system operates on the concept that the system has no knowledge about the content of data provided by users. Therefore, in an implementation of zero-knowledge encryption, a private key, known only to the user, is used to encrypt a given set of data before it is copied to the server, which then manages the encrypted files. In conjunction with other security measures, the use of a key restricts the ability to decrypt the data to the user who originally stored it and creates social sustainability. This paper analyzes the comparative advantages of zero knowledge and traditional security schemes in cloud data storage.

Key Words—Client, Cloud computing, Encryption, Server, Zero-knowledge, Zero-knowledge encryption, Zero-knowledge proof

INTRODUCTION

Over the course of the decade, cloud computing has taken the internet by storm with many major tech companies rolling out cloud services, and in tandem, has also been characterized by a multitude of high profile data leaks.

Cloud computing allows computing to be environmentally and economically sustainable. Sustainability can be defined as, "creat[ing] and maintain[ing] the conditions under which humans and nature can exist in productive harmony to support present and future generations." [1] Sustainability should be considered an obligation of technological development to ensure habitability of our world for future generations.

For cloud computing, sustainability can be defined with environmental, economic, and social considerations. Environmental sustainability is emphasized by an efficient use of resources, while economic sustainability focuses on efficiently meeting the needs of the consumer without endangering the economy itself [2,3]. Social sustainability is the promotion of social stability and equal quality of life [3]. Cloud computing demonstrates each of these properties

Zero-knowledge proofs are not exactly new, but are just starting to gain traction in the computing world. Though complex, the applications of zero-knowledge proofs provide a secure method of communication, especially when coupled with client-side encryption. Zero-knowledge encryption, like cloud computing, is a technology that promotes sustainability, especially economically and socially.

HISTORY OF NETWORK ARCHITECTURE

Mainframes and the Central Computing Model

The landscape of computing has greatly changed since the introduction of commercial computer systems in the early 1950s. This paradigm shift, primarily characterized by the movement away from large, centralized time-sharing systems to a combination of remote processing and interconnected networks of single user machines.

The earliest commercial computers, mainframes such as the Universal Automatic Computer (UNIVAC) I, the first commercially available computer system, were, as reported by CNN in 2001 for the 50th anniversary of the UNIVAC's first purchase, at the smallest, "about the size of a one-car garage." [4] Due to the expensive and sessile nature of these machines, they were primarily utilized by business and government as central data repositories and information processing centers. In fact, according to Morgan Huff, a programmer at Eckert-Mauchly in 1950, the UNIVAC "could handle large volumes of data" and "had a pretty fast processing machine… [that] could satisfy the needs of both

commercial and scientific applications," [5] which were major potential selling points for both varieties of enterprise.

Figure 1 [6]
A UNIVAC I in use. These first commercially available computers were very large and expensive.

With the passage of time and the improvement of computational capability it eventually became possible for multiple people to simultaneously access the same machine with the creation of time-sharing operating systems like Unix, developed at Bell Laboratories in the 1970s. According to Roy A. Allan's *A History of the Personal Computer*, a "time sharing computer system is one that interacts with many simultaneous users through a number of remote consoles." [7] These systems allowed for multiple users to share the computational and storage capacity of the machine while also maintaining compartmentalization. Unix's system of permissions and hierarchical file tree was exceptional at this, in which "[e]ach user has a directory of his own files… that cannot be written on by unprivileged programs, so that the system controls the contents of directories." [8] This system serves to organize each user's content into easily identifiable spaces while also allowing users to grant and restrict access to personal files. This combination of mainframes, time-sharing operating systems, and terminals made it possible for these mainframes to act not only as shared data storage and processing centers, but also, through remote consoles, begin to act analogously to the personal workstations that eventually succeeded them.

Rise of Distributed Computing

In contrast to this trend of large central multi-user systems, as computers continued to grow smaller and more powerful, an opposite design philosophy took hold. As computers both decreased in size and cost, and increased in processing power, it became economically feasible for each user to possess an individual computer. Allan writes that it was in the late 1970's that, "the definition [of the term 'personal computer'] evolved to include a price level that was affordable to the average consumer." [7] Before this, the phrase had held the more literal connotations of any such machine, designed for use by an individual, i.e. not time sharing usage, regardless of power.

The ability of personal computers to independently carry out functions previously reserved to mainframes, along with their nature as machines for individuals, vastly altered how and for what computers were used. The ability to carry out calculations locally, while still having the capacity to network with more powerful machines via the internet opened the doors for the cloud computing model that emerged in the late 2000s.

Despite the current trend, this extreme shift in computing has far-reaching implications. While the production of computers has become increasingly more efficient, the processes and supplies used are unsustainable in the long run. The materials and resources used to build computers are finite and difficult to recycle [9,10]. Therefore as the number of computers built increases, the supply of resources dwindles. As more computers are wanted or needed, and as the quantity of resources diminishes, the price of the computers will increase, which will limit availability of computers. For these reasons, the paradigm of powerful personal computers is environmentally and economically unsustainable. As older computers become obsolete, they are often times thrown in landfills, or recycled illegally in unsafe ways in foreign countries [11]. All of these factors make the process of building computers more expensive which will translate to the end user. An environmentally and economically sustainable solution to this is cloud computing.

CLOUD COMPUTING

How does Cloud Computing Work?

Cloud computing has been a significant recent shift in the computing paradigm. In essence, cloud computing is a commercially available implementation of distributed computing on a massive scale. To clarify, cloud computing is formally defined by the National Institute of Standards and Technology (NIST) as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [12] While being an implementation of a distributed system, with a client machine used as an access point to a remote server, in function, cloud computing is reminiscent of the mainframes of the past, with multiple subscribers sharing the computing and storage capabilities of a single infrastructure. This structure presents its own unique

advantages and disadvantages when compared to both the central computing model and personal computing model.

In addition to its general definition of cloud computing, the NIST also distinguishes between three primary service models: Cloud Software as a Service, Cloud Platform as a Service, and Cloud Infrastructure as a Service.

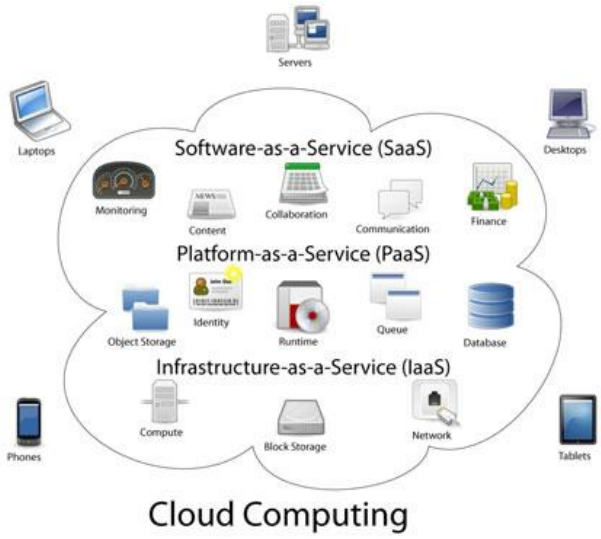

Figure 2 [13]
This is a visual representation of each of the cloud computing models. Each model offers different services which can be accessed on a variety of devices.

Cloud Software as a Service

This model of cloud software as a service (SaaS) entails the ability of the customer to use "the provider's applications running on a cloud infrastructure." [12] SaaS includes any web application where storage or computations occur on the server side, as opposed to, for example, scripts designed to run in the client's web browser. Businesses typically get to use these functions on demand and can share what they choose to with others in their network. Important to note is that the subscriber merely interacts with the provider's applications and has no further control than provided by these applications. Examples of software that would fit well into this model are design software, such as CAD or SolidWorks, and office software, such as Office 365 by Microsoft.

Cloud Platform as a Service

In contrast to the above, cloud platform as a service (PaaS) is the ability for the subscriber to "deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider." [12] PaaS covers cloud computing used as a host or platform for the subscriber's own applications, essentially allowing one party to, for example, access to an account and ability to run software on a remote server. Subscribers to a PaaS cloud provider can customize the application they create, but must use the infrastructure, systems, and data access the cloud provider has to host the application to users. An example of this is Google's App Engine. A subscriber may build an application and it is then executed by Google.

Cloud Infrastructure as a Service

The third classification, according to the NIST's system is cloud infrastructure as a service (IaaS). In IaaS, the customer can "provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications." [12] To summarize, this classification is essentially the ability to rent a machine with near full software customizability and minor hardware customizability. IaaS allows subscribers to be capable of creating nearly anything. The only thing a subscriber cannot do is modify the underlying cloud infrastructure. An example of an IaaS cloud service is Microsoft Azure, where subscribers can create and modify applications and services for consumers.

These three models form a continuum of the amount of control the subscriber has over the machine provided by the service. IaaS gives the user the most control and variety, but not all businesses need or possess the skills to build from this. At the other end of the spectrum, SaaS is not customizable, but is very useful to businesses targeting certain software packages. Cloud computing is befitting of many businesses and gives subscribers many options. However, common to each model and key to the concept of cloud computing is that despite the varying control over and interact with the systems and applications running in the "underlying cloud infrastructure" [12], in each case, it is the provider who manages, controls and maintains this infrastructure.

Applications of Cloud Computing

Utilization of a cloud-based IT infrastructure provides numerous potential benefits to a subscriber over a more traditional central or unit-wise system. In particular, Professor Jaydip Sen, a network technology specialist at the Praxis Business School in Kolkata, India, recognizes three particular economic advantages. The first, "[t]he illusion of infinite computing resources available on demand" [14] ensuring that it is unnecessary "to plan far ahead for provisioning" [14], freeing up the subscriber to quickly adjust their computational capabilities on demand and reducing the risk of acquiring an inappropriate number of resources. The second, the "elimination of an up-front [monetary] commitment by cloud users" [14] negates the financial burden of the expenses otherwise posed by hardware acquisition. Third and final, "[t]he ability to pay for use of computing resources on a short-

term basis… and release them when the resources are not needed "[14] eliminates the need to keep and maintain unused equipment. Therefore, cloud computing is economically sustainable because it allows for an impartial and efficient allocation of computing resources, as well as environmentally sustainable in that it reduces the physical waste that is unused hardware.

In addition to cost effectiveness, another advantage offered by cloud computing is convenience. By definition, cloud computing takes the form of remotely accessible applications, platforms, and infrastructure, and as such, due to its decentralized nature, can be accessed from any location with an internet connection, simultaneously by multiple clients. This is analogous to the time-sharing mainframes of the past, with any number of personal computers serving similarly to console access points. However, unlike these mainframes, cloud infrastructures are not centralized, and can span multiple machines connected to the same network. This allows for mobile storage and retrieval of data and even collaboration between users in some applications, rather than access to these resources being limited to those physically present at the hosting site(s).

In summary, cloud computing offloads the costs associated with acquiring and maintaining a traditional computing infrastructure while also promoting efficient use of resources though the ability to flexibly reallocate computing capacity as needed. This efficient use of resources allows large scale computing to be environmentally and economically sustainable. Additionally, the decentralized and net based nature of cloud infrastructure allows it to be simultaneously remotely accessed by any number of network connected clients in multiple locations.

Vulnerabilities of Cloud Computing

Despite the convenience and economic advantages of cloud computing, the distributed structure gives many businesses cause for concern. As more businesses utilize cloud services, this concern grows. One of the biggest threats is the sheer size of the cloud computing network. The large size of cloud infrastructure would give anyone working for the provider access to many different points of entry [15]. It is difficult to keep surveillance on all aspects of infrastructure simultaneously and a great deal of trust is given to employees. Besides the large infrastructure, the number of users accessing the cloud network is also cause for concern. Cloud computing services containing thousands of users are typically made up of that many domains. If one of these domains is breached, it is a threat to all other domains. Accessing a cloud computing network is not difficult for those trying to launch an attack. These services are provided to users via the Internet, which everyone has access to. A breach in a user's account while signing in or through a scam could give an attacker unwarranted access. Since data is being uploaded to remote servers, anyone who can access these servers can thereby access user data.

Cloud service providers have gone to great lengths to assure users that their data is safe, but can they be trusted? Companies have instituted many regulations concerning how and where hardware is stored and record keeping practices for those who have access, but is that enough [16]? Instead of having to rely on public policy, users should be able to rely on something more trustworthy and straight-forward. Protection of user data is paramount. The best solution to cloud computing vulnerabilities is zero-knowledge encryption.

With an ever growing societal dependence on computer networks, society becomes increasingly vulnerable to damage wreaked through destabilization of these systems. If a cloud system is compromised, subscribers to that service, both individual users and businesses, are exposed to any harm incurred to or though that system. This presents a destabilizing effect on the quality of life of users and businesses making use of system, and through these on society and the economy at large. Thus, the security provided by zero knowledge encryption protects subscribers and promotes social and economic sustainability through trust between users and providers of cloud services.

ZERO KNOWLEDGE ENCRYPTION AS A CLOUD SOLUTION

History of Zero Knowledge Proofs

Zero knowledge proofs were first defined by Goldwasser, Micali, and Rackoff in 1985 [17]. Their paper addresses the fact that many proofs require certain information in order to "verify the correctness of a statement." [17] This information is enough to verify the given proof, but is, "usually much more." [17] This leads to, "How much knowledge should be communicated for proving a theorem… ?" [17] Why should we have to give any sensitive information for the purpose of validation in data storage? If the purpose of encryption is to protect our data from others, it is counterintuitive to reveal said data. However, it is important to note that zero-knowledge is not analogous to zero-information. Zero-knowledge proofs still require information to be relayed. What is important is the fact that the information being relayed does not give the verifier or observing parties any knowledge in a way that can be used to reconstruct what is being proven.

Formally, a zero-knowledge proof is defined by three conditions, including complete, sound, and perfect (also known as zero-knowledge) [18]. A proof must meet these three conditions totally in order to be considered zero-knowledge. A proof is considered to be complete if the person verifying agrees at every stage something is being verified and is correct. The second condition, sound, is met if the person verifying the information is sure that the information is correct with a high probability. Finally, the proof is considered to be perfect or zero-knowledge if the

person verifying the information has gained no pertinent knowledge about that which they are verifying. In order to illustrate these abstract concepts, the following example has been constructed.

Example Proof Using a Sudoku Puzzle

A proof is said to be zero knowledge so long as it satisfies the conditions of being complete, perfect, and sound. Consequently, any method, algorithm, or process that complies with these conditions can be considered zero knowledge, regardless of form or complexity. With this in mind, it is useful to provide a practical example to aid in the explanation of the definition of a zero knowledge proofs, especially in relation to these three conditions. To this end, we have elected to use the following scenario, utilizing a Sudoku puzzle as a medium.

Sudoku is a type of logic puzzle which features a nine row by nine column grid filled with the digits one through nine. This grid contains, in total, 81 cells subdivided into nine three by three subsections each containing nine cells. The object of the puzzle is to fill each of the rows, columns, and subsections with the digits 1 to 9 such that no digit occurs more than once in any one row, column, or subsection. Usually, some digits are already filled out for a given puzzle, and it is the job of the puzzle solver to use these predetermined digits to infer the contents of the empty cells.

Suppose a hypothetical person and a friend are asked by a third party to solve a Sudoku puzzle. After working for a time, the person informs their friend that they have found the solution. The friend insists that they backup their claim. The person could show their solution, however, the friend has not yet finished and doing so would give them the answer to the puzzle. Therefore, the answer needs to be proven in a way that shows a solution has been figured out without giving away the solution itself, i.e. a zero-knowledge proof.

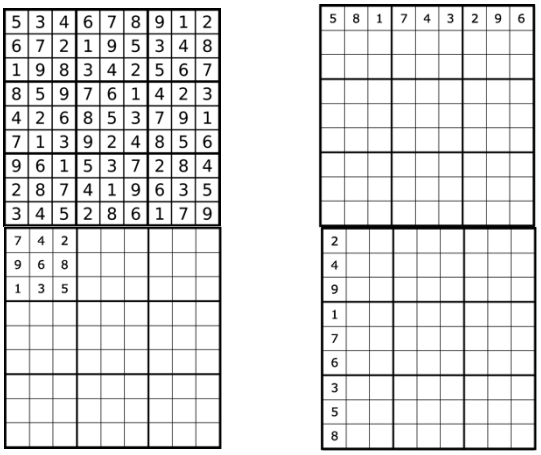

Figure 3 [19]
A visual representation of what is being viewed in the Sudoku proof.

The solution to the problem is as follows. First, the person replaces each of the digits with another, unique, digit. For example, all twos become threes, all threes become nines, etc. Then, they allow the friend to select one row, column, or subsection, and then show them these cells and only these cells, filled with substituted values. If the friend sees that there are no repeated digits, they know that this solution for that particular set of cells is valid. This process is repeated, substituting new values and having the friend select a new row, column, or subsection each time (see figure 1 for clarification). The more valid rows, columns, and subsections they see, the more confident the friend can be that the entire solution is valid. After viewing each row, column, or subsection, it is obvious the person has completed the puzzle, but the friend has no knowledge of how the puzzle is solved.

This proof is complete in that each individual row, column, and subsection can be individually determined to be valid according to the rules of Sudoku, sound in that after a large number of valid iterations, the probability of the entire solution being invalid becomes negligible, i.e. the validity of the solution cannot be faked, and perfect in that it is impossible for the friend to reproduce the solution from the information they are given.

Encryption and the Discrete Logarithm Problem

A logarithm is the power to which a base needs to be raised in order to get a number. For example, the $\log_2 4 = 2$ since 2^2 is 4. In a general form, this can be written as $\log_a b = x$ and $a^x = b$. The only numbers considered are positive integers, which are numbers that are non-decimal and non-fraction. Given a and b, solving for x is trivial for certain numbers, but this does not hold true for all sets of numbers. Given two sets of numbers, a and a set b that exists within a, finding a number x that satisfies $a^x = b$ is called the discrete logarithm. Finding the discrete logarithm over specific sets of numbers can become extremely difficult and time consuming, especially in repetitive groups [20]. Current technology is not capable of efficiently solving for these discrete logarithms.

In cryptography, the Discrete Logarithm Problem is the foundation for many encryption algorithms. Key based encryption systems heavily rely on this concept. Although solving the discrete logarithm problem would render most current encryption techniques defunct, researchers have worked to solve this problem for decades. The simplest way to solve this problem would be a prime factorization. Primes are numbers whose only factors are one and itself. For example, the prime factorization of 24 is 2, 2, 2, and 3. Multiplying these numbers results in 24, and they cannot be broken down further into smaller numbers. 2 and 3 are prime numbers since 2 is equal only to 2 times 1 and 3 is equal only to 3 times 1. However, as the size of the number increases, the time it takes to complete a prime factorization can grow exponentially. The size of the numbers used for encryption would take thousands of years to factor by hand. In order to

get around this, scientists have tried creating different algorithms based on factorization to speed up this process.

There have been many algorithms proposed over the years. Old algorithms have been improved upon and new algorithms created. Although none are established as a solution to the discrete logarithm problem, some have garnered attention. Two of the most well-known of these algorithms are the General Number Field Sieve and Shor's Algorithm.

The General Number Field Sieve

$$O\left\{\exp\left[c\,(\log n)^{1/3}\,(\log\log n)^{2/3}\right]\right\},$$

Figure 4 [21]
The GNFS algorithm. This algorithm is used to calculate prime factorizations.

The General Number Field Sieve (GNFS) is an algorithm for calculating the prime factorization of large numbers. The GNFS was developed by mathematician John Pollard. Using this method for general numbers, the GNFS is a very fast and powerful algorithm. [21,22] However, it is not very suitable to find the prime factorizations of the numbers used in encryption. A better description of the GNFS, when compared to other methods and algorithms, is reasonable rather than fast.

During the 1990's, RSA Security, a computer company, created the RSA Factoring Challenge. The Factoring Challenge consisted of a list of numbers that were the product of multiplying two large prime numbers together. RSA Security encouraged people to try solving these difficult problems and offered rewards for some. The factorization of RSA-768, a 232-digit long number (which is 768 bits), was completed in two years. However, this time frame is slightly misleading. RSA-768 was not factored by a single computer, but actually hundreds. In reality, the scientists who completed this project estimate that if this calculation was done on a single computer, it would have taken around 1500 years, or possibly longer. [23] Therefore, using this method to decipher an encryption key would be extremely impractical.

The GNFS, and other similar factorization algorithms, can take absurd amounts of time to solve problems like RSA-768 because they do not operate in polynomial time. Polynomial time is considered to be fast, especially compared to something like exponential time. The time it takes to complete an algorithm is considered polynomial time if the maximum time is a function in the form of a polynomial. For example, if the algorithm takes twice as long to complete as the number of inputs and there are two inputs, then the time is four. For an algorithm in exponential time, something with only two inputs could result in a vastly larger amount of time.

For these reasons, encryption utilizing the discrete logarithm problem is very safe and reliable in the foreseeable future.

Shor's Algorithm

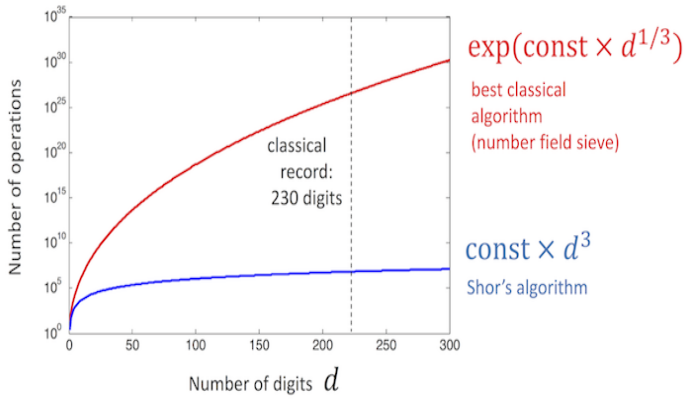

Figure 5 [24]
This graph shows the comparison of the number of digits to the number of operations for the GNFS and Shor's Algorithm. This graph shows how the time to calculate the GNFS increases exponentially, and how Shor's Algorithm is much faster.

Since the conception of quantum computing, scientists have struggled to employ this technology. Quantum computation is based on the concept of qubits, rather than bits, like a classical computer. Bits, are made up of zeros and ones, while qubits are made up of zeros, ones, and a combination of both. This combination of zero and one is called superposition. This altered state allows for a more quick and efficient calculation compared to a classical computer. Quantum computers are an excellent concept and likely will be very beneficial to mankind, but they are currently unattainable to make. Very simple quantum computers have been created, but they are highly impractical and difficult to construct.

Nevertheless, some scientists have worked extensively to create algorithms to run on quantum computers. Most notably, Shor's Algorithm has attracted substantial attention.

Shor's algorithm was created by Peter Shor at Bell Laboratories in 1994. After several popular papers on quantum computers were published, Shor started working on algorithms for solving for the discrete logarithm and factoring. When using Shor's algorithm on a quantum computer compared to the GNFS on classical computer, the times to solve the same problem are not remotely comparable. What would take a classical computer thousands of years could be done in less than a second on a quantum computer [25].

Since Shor's algorithm with quantum computation is so powerful, many current encryption techniques could be cracked using it. However, advancements to make this sort of technology practical are a long time away. Therefore,

encryption is a dependable and trustworthy method of keeping data safe.

Integration of Zero Knowledge Encryption and Cloud Computing

On a cloud computing network, there may be thousands of clients. All of these clients are uploading files and retrieving them when needed. If all of this data is stored together on one server then what is keeping someone from taking another user's data or leaking everyone's data? The short answer is not much. Cloud computing is vulnerable to attacks from many different angles, from rogue employees to random hackers. A solution to this is a combination of Proof of Data Possession (PDP) protocols and zero-knowledge encryption.

With the arrival of cloud computing, users wanted a way to quickly and efficiently verify their files which the cloud network was holding. In the case of an unreliable provider, users needed to ensure that their data was complete, unaltered, and secure. From this, PDP protocols were created [26]. PDP protocols allow the user to verify their files on the cloud network without actually retrieving them. In doing this, no files are transferred out of the cloud and the user is able to verify the existence and correctness of their data.

One problem with traditional PDP protocols is that they use metadata in order to prove to the user that their files are uncorrupted. This means that the cloud provider is accessing user data in order to validate to the user. Cloud providers should not have to go through this process. Instead, users and cloud providers should be interacting through a zero-knowledge protocol [27].

First, any files a user uploads to a cloud computing network should be encrypted. Current encryption methods are extremely safe and reliable. By utilizing encryption, the user is ensuring the that their files are completely and utterly protected. This process can be completed by using a third-party application to get a public and private key for encrypting and decrypting user files. Under no circumstances should a cloud computing provider also provide encryption of user data. If the provider is giving out private and public keys, that means they have access to the keys and may store them, therefore rendering the encryption useless.

Once the files are encrypted, they may then be uploaded to the cloud network. The cloud network is where data is most susceptible to attack. It is important to note that even if the cloud network is breached and user data is leaked, as long as data is encrypted and the cloud service does not have user keys, there is not a large threat to the user due to the complexity of the discrete logarithm problem.

Now as a user, one would like to retrieve said files. PDP protocols were put into place to allow the user to verify the authenticity of their files, but it would also be good practice to follow a PDP protocol to retrieve files. Additionally, to make the process even safer, the cloud computing network should verify that the user is actually the person they claim to be, lest a hacker obtain access to a user's personal computer. An efficient, secure way for the user to verify their identity and the cloud to verify its identity is through a zero-knowledge proof.

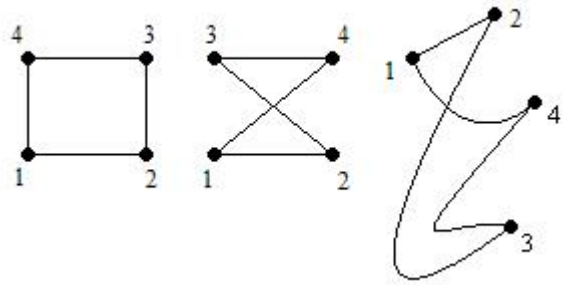

Figure 6 [28]
A simple example of two isomorphic graphs. The left and center graphs show two graphs, while the graph on the right shows the isomorphism between the two.

There are many ways the user and cloud could do this, but a simple, yet effective way is through graph isomorphisms. First, the graphs in this proof are defined as having thousands of points, known as vertices, and lines connecting certain vertices, known as edges. Isomorphic graphs are defined as graphs containing the same vertices and edges. Though proving graph isomorphism may seem like an easy task, it is notoriously difficult, and in some cases nearly impossible. The cloud network and user utilize a program to create a complex graph and a mechanism to interact with each other when logging into the cloud computing network. Initially, each program will create a unique graph and isomorphism of their respective graphs. To clarify, graphs G_1 and G_2 are created and some process f when applied to G_1 results in G_2. Each program will create a random permutation of one of the graphs, denoted by $\pi G=G^*$. Each permuted graph is sent to the other and stored for later use. When the user would like to retrieve their files, the program will prompt the cloud network for a random integer, either one or two. The program then returns one of these numbers. If the cloud network returns one, the opposite of the permutation π is performed on the graph G^* and the program is given G_1. If the program returns two, the process f is applied to the permutation π and then is performed on G^*. The program then returns G_2 to the cloud network. The program on the cloud network can then verify that the graph G^* is an isomorphism of G_1 without knowing the process f which results in G_2. This same process is then performed again, only instead the cloud network program sends its permuted graph to the user. Once both programs verify the identity of each other using this example, the user and cloud network can confirm that both are who they claim to be. After this process,

the programs create new permutations and send them to each other for the next log in.

This example is a zero-knowledge process because it fulfills the three qualifications of soundness, completeness, and perfectness. The proof is complete because at each point the verifier can see that each vertex and edge is isomorphic to the other. The proof is sound because graph isomorphism is very complex to find and can not be faked. Therefore the verifier is certain to a high probability the prover is sharing true information. The proof is perfect because no information about the true isomorphism between the graphs is shown, only an isomorphism between one of the graphs and a different graph. Zero-knowledge encryption promotes social sustainability in cloud computing because it fosters a stable network for all users where all data is safe. The social sustainability of zero-knowledge encryption fosters economic stability because if all people's data is objectively stable, the markets involved with zero-knowledge encryption will remain stable due to trust in the technology.

Sustainability for the Future

Cloud computing coupled with zero-knowledge encryption creates a sustainable computing platform for the future. Cloud computing fosters environmental sustainability through its centralized nature. The EPA estimates that 20 to 50 metric tons of E-waste, including computers and similar technological devices, are thrown away each year [29]. Sales of computers and laptops in the United States alone reached 61.1 million in 2013 [29]. Sales are projected to continually rise in the long run, especially in compact and mobile computer devices [29]. As the amount of computers made and purchased grows, resources will dwindle and pollutants will be created as cheap, unregulated recycling processes are used to keep up with demand. Estimates vary, but around 81% of the energy consumed by computers is used in their creation [29]. "To create one computer and monitor, it takes 530 pounds of fossil fuels, 48 pounds of chemicals, and 1.5 tons of water," [29] which create pollutants and emissions in their own right. Undoubtedly, the trends in computer production are environmentally and economically unsustainable due to the depletion of resources and the resulting skyrocket in computer pricing. If all people would progressively convert to cloud platform based computation, resources would be conserved and therefore create a more sound economic market in computation. The economic sustainability of cloud computing allows for social sustainability due to the fair market which creates equal opportunities for all people to use cloud computing.

Zero-knowledge encryption creates a stable computing environment where all people's data is securely protected regardless of what services they pay for while creating trust between providers and subscribers of cloud based services. This confidence in security is itself a form of social sustainability. Zero-knowledge encryption also creates economic sustainability by protecting data from theft and leakage and any accompanying harm it might cause to businesses.

THE BIG PICTURE

The paradigm shift toward cloud computing since the late 2000s is a symbolic return to the centralized computing infrastructure of the mid to late 1900's. While the mainframes of the past would often be owned and operated by the companies and organizations who had purchased them, cloud infrastructure is provided by an outside organization. Cloud computing is inherently multi-user and mobily accessible. Cloud services often provide more variable levels of control over software configuration by the subscriber. Despite this variable level of control, under each classification the underlying hardware is provided and maintained by the provider, effectively allowing a subscriber to utilize cloud computing as a means to eliminate the heavy costs associated with acquiring and managing hardware in an economically sustainable manner, while being environmentally sustainable by reducing the strain on resources imposed by this extra hardware.

Although cloud computing offers many advantages over traditional IT infrastructure, its distributed structure coupled with the sheer size and complexity of most cloud systems make it nearly impossible to efficiently surveil for threats. The interconnectedness of these systems in tandem with the large quantity of traffic they receive present ample opportunity for hackers or even disgruntled employees to gain access to subscriber data across the system. Zero knowledge encryption presents a solution to this dilemma that removes the need for the user to blindly trust in the service provider to maintain the integrity of the back-end, increasing the social and economic sustainability of cloud-based systems through this security.

Zero knowledge encryption is the joined application of the security concepts of zero knowledge proofs and encryption. Zero knowledge proofs are a means of proving one's possession of given knowledge without revealing any information to a verifier that can be used to reconstruct this knowledge. A zero knowledge proof is defined by the principles of completeness, that the pieces of revealed information can be independently verified by the verifier, soundness, that as the quantity of verified information approaches the complexity of the original problem, the probability of the solution being valid approaches one hundred percent, and perfectness or zero knowledgeness, that the verifier cannot reconstruct the original knowledge from the revealed information.

Encryption is a means of obfuscating information that makes use of the discrete logarithm problem. These are calculations that are easy to compute but nearly impossible to solve in the reverse direction without knowing the factor, known as a key, used to originally compute them.

An implementation of zero knowledge encryption would utilize client-side encryption in order to ensure that

only the subscriber possesses the key used to encrypt their data, and therefore, due to the complexity of the discrete log problem, the ability to decrypt it. Encrypted file retrieval would make use of zero knowledge proofs in verifying file integrity and user and cloud identity.

Cloud computing creates environmental and economic sustainability through more efficient use of resources than personal computers, and zero-knowledge encryption creates social and economic sustainability by protecting subscriber data.

SOURCES

[1] United States Government. US EPA. "Learn About Sustainability." EPA Website. 10.16.2016. Accessed 3.28.2018. https://www.epa.gov/sustainability/learn-about-sustainability#what

[2] J. Morelli. "Environmental Sustainability: A Definition for Environmental Professionals." Journal of Environmental Sustainability. 11.2011. Accessed 03.28.2018. http://www.environmentalmanager.org/wp-content/uploads/2011/09/Article2Morelli1.pdf

[3] A.D. Basiago. "Economic, Social, and Environmental Sustainability in Development Theory and Urban Development Practice." Environmentalist. 06.1998. Accessed 03.28.2018. https://www.amherst.edu/system/files/media/0972/fulltext.pdf

[4] "50th anniversary of the UNIVAC I" CNN. 06.14.2001 Accessed 02.26.2018. http://www.cnn.com/2001/TECH/industry/06/14/computing.anniversary/

[5] E. Adams et al. "UNIVAC Conference." Charles Babbage Institute. 18.05.1990 Accessed 02.26.2018. https://conservancy.umn.edu/bitstream/handle/11299/104288/oh200uc.pdf?sequence=1&isAllowed=y

[6] This photo of the original UNIVAC I was taken from an article from Time Magazine's website. Time Magazine credits the photo to Underwood Archives, Getty Images.
M. Fabry. "The Story Behind America's First Commercial Computer." Time Magazine. 03.31.2016. Accessed 03.28.2018. http://time.com/4271506/census-bureau-computer-history/

[7] R. Allan. "A History of the Personal Computer." Allan Publishing. 10.01.2001. Accessed 03.01.2018. https://archive.org/details/A_History_of_the_Personal_Computer

[8] D. Ritchie, K. Thompson. "The UNIX time-sharing system." Communications of the ACM. 07.1974. Accessed 02.27.2018. https://dl.acm.org/citation.cfm?id=361061

[9] K. Razi. "Resourceful Recycling Process of Waste Desktop Computers: A Review Study." Resources, Conservation, and Recycling. 07.2016. Accessed 03.28.2018. https://www.sciencedirect.com/science/article/pii/S0921344916300519

[10] United States Government. US EPA. "Electronics Donation and Recycling." EPA Website. 11.07.2017. Accessed 03.28.2018. https://www.epa.gov/recycle/electronics-donation-and-recycling

[11] Greenpeace. "Where Does E-Waste End Up?" Greenpeace.org. 02.24.2009. Accessed 03.28.2018. https://www.greenpeace.org/archive-international/en/campaigns/detox/electronics/the-e-waste-problem/where-does-e-waste-end-up/

[12] L. Badger, T. Grace, R. Patt-Corner, J. Voas. "DRAFT Cloud Computing Synopsis and Recommendations." National Institute of Standards and Technology. 05.2011 Accessed 02.27.2018. https://permanent.access.gpo.gov/gpo28772/Draft-NIST-SP800-146.pdf

[13] This photo gives a visual representation of each cloud computing service model. The photo is from an article on cloud computing from dummies.com.
K. Withee, J. Reed. "Where Office 365 Cloud Computing Is Today." Dummies.com. Accessed 03.28.2018. http://www.dummies.com/software/microsoft-office/office-web-apps/where-office-365-cloud-computing-is-today/

[14] J. Sen. "Security and Privacy Issues in Cloud Computing." Cornell University Library. 03.20.2018. Accessed 02.11.2018. https://arxiv.org/pdf/1303.4814.pdf

[15] A. Girma, M. Garuba, J. Li. "Analysis of Vulnerabilities of Cloud Computing Environment Service Models and Its Main Characteristics." IEEE. 01.06.2015. Accessed 03.01.2018 http://ieeexplore.ieee.org.pitt.idm.oclc.org/document/7113474/

[16] N. Santos, K. Gummadi, R. Rodrigues. "Towards Trusted Cloud Computing". USENIX. 2009. Accessed 1.15.2018. https://www.usenix.org/legacy/event/hotcloud09/tech/full_papers/santos.pdf

[17] S. Goldwasser, S. Micali, C. Rackoff. "The Knowledge Complexity of Interactive Proof Systems". ACM Digital Library. 1985. Accessed 1.28.2018. https://dl-acm-org.pitt.idm.oclc.org/citation.cfm?doid=22145.22178.

[18] O. Goldreich, S. Micali, A. Wigderson. "Proofs that Yield Nothing But Their Validity All Languages in NP Have Zero-Knowledge Proof Systems." Journal of the Association for Computing Machinery. 07.01.1991. Accessed 01.15.2018 https://dl.acm.org/citation.cfm?doid=116825.116852

[19] All Sudoku puzzles were created and printed from the following website. https://sudoku9x9.com/

[20] K. McCurley. "The Discrete Logarithm Problem." American Mathematical Society. 1990. Accessed 02.26.2018. http://www.mccurley.org/papers/dlog.pdf

[21] E. Weisstein. "Number Field Sieve." MathWorld-A Wolfram Web Resource. Accessed 02.26.2018. http://mathworld.wolfram.com/NumberFieldSieve.html

[22] A. Lenstra, H. Lenstra, M. Manasse, J. Pollard. "The Number Field Sieve." Mathematics Subject Classification. 1991. Accessed 02.26.2018. https://www.math.leidenuniv.nl/~hwl/PUBLICATIONS/1993d/art.pdf

[23] T. Kleinjung et al, "Factorization of a 768-Bit RSA Modulus." Advances in Cryptology – CRYPTO 2010. 2010. Accessed 02.26.2018. https://link.springer.com/chapter/10.1007/978-3-642-14623-7_18

[24] This photo was taken from the website qiskit.org, which is part of IBM Research and the IBM QX Team.
IBM. "Shor's Algorithm." 2017. Accessed 03.28.2018. https://www.qiskit.org/ibmqx-user-guides/full-user-guide/004-Quantum_Algorithms/110-Shor's_algorithm.html

[25] P. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring." IEEE. 1994. Accessed 02.26.2018. http://ieeexplore.ieee.org.pitt.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=365700

[26] N. Kaaniche, M. Laurent. "SHoPS: Set Homomorphic Proof of Data Possession Scheme in Cloud Storage Applications" IEEE. 07.02.2015 Accessed http://ieeexplore.ieee.org.pitt.idm.oclc.org/xpls/icp.jsp?arnumber=7196517

[27] N. Kaaniche, E. Moustaine, M. Laurent. "A Novel Zero-Knowledge Scheme for Proof of Data Possession in Cloud Storage Applications." IEEE. 08.06.2014. Accessed http://ieeexplore.ieee.org.pitt.idm.oclc.org/document/6846488/

[28] This picture is credited to Sudev Naduvath of the Vidya Academy of Science and Technology. It was posted by Naduvath to the website researchgate.net. https://www.researchgate.net/post/What_is_graph_isomorphism

[29] Electronics Takeback Coalition. "Facts and Figures on E-Waste and Recycling." Electronics Takeback Coalition. 06.26.2014. Accessed 03.29.2018. http://www.electronicstakeback.com/wp-content/uploads/Facts_and_Figures_on_EWaste_and_Recycling.pdf

SOURCES CONSULTED

V. Chang, Y. Kuo, M. Ramachandran. "Cloud computing adoption framework: A security framework for business clouds." Future Generation Computer Systems. 31.9.2015. Accessed 1.15.2018. https://www.sciencedirect.com/science/article/pii/S0167739X15003118

H. Dinh, C. Lee, D. Niyato, P. Wang. "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches." Wireless Communications and Mobile Computing. 10.11.2011. Accessed 02.11.2018. https://pdfs.semanticscholar.org/7f6f/23fadaeec2f2777e7234efae3d66ad4245a8.pdf

S. Hamdi, F. Mahmud, S. Zuhori. "A Compare between Shor's quantum factoring algorithm and General Number Field Sieve." IEEE. 09.10.2014 Accessed 02.11.2018. http://ieeexplore.ieee.org.pitt.idm.oclc.org/document/6919115/

H. Lipmaa. "Zero Knowledge and Some Applications." Helsinki University of Technology. 05.16.2004. Accessed 02.26.2018. http://kodu.ut.ee/~lipmaa/teaching/Bergen2004.pdf

B. Lynn. "Zero-Knowledge Proof Systems" crypto.stanford.edu Accessed. 1.15.2018 https://crypto.stanford.edu/pbc/notes/crypto/zk.html

C. Yang, M. Zhang, Q. Jiang, J. Zhang, D. Li, J. Ma, J. Ren. "Zero knowledge based client side deduplication for encrypted files of secure cloud storage in smart cities." Pervasive and Mobile Computing. 03.14.2017. Accessed 1.15.2018. https://www.sciencedirect.com/science/article/pii/S1574119217301669

T. Hales. Interview on concepts in graph theory, zero-knowledge proofs, and mathematical cryptography. University of Pittsburgh Department of Mathematics. 02.21.2018.

J. Wheeler. Interview on mathematical concepts used in encryption. 02.22.2018.

J. Wheeler. Interview on mathematical concepts used in encryption. 03.01.2018.

ACKNOWLEDGEMENTS

Thank you to our co-chair, Samuel Birus, our chair, Greg Wunderley, and our writing instructor, Rachel McTernan, for their guidance, feedback, and advice to make this paper the best it could be. Thank you to Dr. Thomas Hales of the University of Pittsburgh Department of Mathematics for explaining some concepts of graph theory and introducing us to some basic proofs of zero-knowledge protocols. Last, but certainly not least, a big thank you goes to Dr. Jeffrey Wheeler of the University of Pittsburgh Department of Mathematics for his help breaking down complex mathematical concepts into something we could understand and providing valuable resources to further our research.