Contents

About this document

This document gives a comprehensive discussion of my current academic interests. They are categorized into broad fields as sections, and each subsection, describing my interest for a subfield of it, will generally give my motivation for studying that field and usually end with a current open problem that I aim to solve during my research in that subfield.

If you are a professor, I might have sent the whole, a part, or a modified version of this document to you, as my statement of interest. Generally, you only need to read the subfields that I want to be working in with you; see the above table of contents. If you have time, however, I still recommend a linear read through the document to see the whole picture of my ambition and passion. Finally, to get the latest version of the document, go to `https://github.com/guanyuming-he/guanyuming-he/blob/main/interests/main.pdf`.

Part I

My academic interests

1 Security: Introduction into my interests

I believe security is a good entrance into my current interests. It awards both attention to detail and high-level systematic view; its subfields connects directly with both theoretical and practical areas that fascinates me, and it offers effective tools to solve the imminent problems I face in real life. This section will take such a flow that the subfields of my interest will be first led to from real world problems and motivation, and then be explored with my more academical and philosophical pursuits.

The term *security* in computer science generally carries the sense of achieving *security goals* through *mechanisms or properties* of a system, despite the presence of adversaries in a *threat model*. The threat model has to be carefully chosen and reviewed; there would be no security against an omnipotent adversary.

Unlike in math, one cannot arbitrarily wiggle the threat model to one's desire. On the contrary, one has little control to how one's digital rights are treated, which are increasingly threatened (e.g. [34, 83, 73]). A more prominent example is the continuous push for exceptional access to people's data and communication in various forms by governments[1], ignoring expert opinions against such measures [2, 1, 5]. Prof. Anderson has frankly described the fight for the control of cryptography (thus a large portion of peoples' digital rights) between governments and all who try to fight back with the term "(thirty years of) crypto war" [6]. In this section, you will see a similar tone to Prof. Anderson's honesty which I admire — I aim to approach my work with the same honesty — to confront problems as they are, to fight against injustice even when inconvenient, and to maintain my integrity throughout my academic journey. In fact, Prof. Anderson's section *Privacy and freedom issues* [6] on his homepage was one major inspiration that set me on the path of cybersecurity, but which I will not elaborate in the document.

Moreover, I unfortunately have lived in a place where peoples' digital rights are much more aggressively invaded, particularly in the form of systematic, comprehensive, and far-reaching censorship and surveillance [7] [91, Sect. 5]. These threats present one of the major real world problems I aim to address by my research in security and my main realistic motivation. Through the document, you will see my philosophical motivations as well.

[1] For a recent instance, EU revived the Chat Control proposal in 2025. See `https://eutechloop.com/time-is-running-chat-control/`.

1.1 Secure Systems: Understanding the mechanisms

1.1.1 Low-level system details

Beside the threat model, when we talk about security, we also implicitly assume another model, environment, or host, that encapsulates the problem. For instance, in the context of isolation,

- The host that encapsulates process isolation is the operating system kernel.
- The host that encapsulates virtual machine isolation is the hypervisor, often including hardware support.
- The host that encapsulates air-gapped machine isolation is the physical world, or more abstractly, the physical laws.

These hosts form a hierarchy. For instance, processes inside an OS may run inside a virtual machine, on hardware, within the physical world.

As a result, a security researcher has to understand these hosts, each to a different degree, depending on how secure one wants one's systems to be. Even if a system is formally verified to be secure within a host, it can still be attacked from interactions with an outer host. That is well examplified by the Meltdown attack, which breaks both kernel and hypervisor's isolation of memory, because an outer host, the hardware, has a vulnerability [53]. In fact, Meltdown touches more than logical flaws in hardware; it relies on the timing difference of cache accesses to extract information from it [92], one that directly connects with the physical world. At this point, I feel that I should also mention the Spectre attacks family [50], which, unlike Meltdown, does not exploit a clear logic bug or fault in the hardware. Instead, it work by observing the side effects of various implementations of speculative execution, which is arguably not a bug but simply a feature implemented without caution to how information may be leaked via the outer hosts.

Indeed, there is little hope to understand and model the universe thoroughly and accurately in one's life-time, and one can expect endless discoveries of how an otherwise secure system can leak information in unexpected ways, such as power and radiation via physical effects, which opens to a large range of attacks [51, 11, 32, 92], collectively defined as side-channel attacks. Then, one must ask, is it futile trying to cope with every such possible attack? My opinion is that it depends on the current understanding of the whole cybersecurity community. One novel attack just discovered may be increasingly understood and gain popularity by the whole community and thus gradually become a necessary part every secure system designer should take into account. I view this as an everlasting dynamics where people continuously push the boundary of our understanding, and consequently expanding the requirements for all in the community.

Therefore, a good understanding of the most common and well-understood hosts, the operating systems and the hardware (including the architecture), will be essential for me as a security researcher. Additionally, figuring out how to perform clever hacks (as in the hacking culture of MIT, not the mainstream meaning of cracking) based on knowledge of details of a system gives a great sense of achievement to me. Moreover, this continuous expansion of knowledge frontier carries a special implication in security. Whereas a mathematician is perfectly fine to specialize in one direction without knowing every details in other fields, a computer system can never be secure if only designed with one perspective. For now, a good system is usually designed by collaboration; and it will be a very interesting challenge to figure out how one single researcher can be empowered to do this task better; could the LLMs be a potential assistant who fills the knowledge gaps? Finally, this is one major place where cybersecurity connects directly with a broad range of other fields, including software engineering, systems architecture, networking, and physics. These together contribute to my interest in the low-level details of secure systems.

1.1.2 A high-level view of the Internet

A high-level understanding of how the digital infrastructures work is essential to understand what enables the threats to digital rights on top of them. The current networking infrastructure, in my opinion, has conflicting properties. On one hand, the fundamental problem of the impossibility to link every two computers requires sharing of links, a solution that welcomes centralization. On the other hand, the poor scalability of simple sharing schemes of a link (usually via a switch) necessitates a better scheme to extend a small network globally. To achieve this, the Internet relies heavily on delegation and distributed structures, such as topological divisions of its address space and delegation of most routing responsibilities to each individual networks (e.g. ISPs). Its BGP, opearing in between them, is mainly concerned exchanging reachability information, whereas its IGPs handle routing paths and allow different networks to implement different routing policies.

Therefore, the Internet has become a mixture of centralization and decentralization, where each end node is managed by an ISP network, yet no single ISP runs the whole Internet. Perhaps surprisingly, I found this hybrid structure more optimal for localized sabotage at the state level, creating "sub-Internet"s, each of which is crippled at a different level.

Unfortunately, the lower layers of the Internet have proven to have a great inertia for change. Handley gave a nice discussion on it in 2006 [41], and the trend he described has mostly been the same since then: "*the core Internet protocols have not changed significantly in more than a decade, in spite of exponential growth in the number of Internet users and the speed of the fastest links.*" [41]. On the other hand, the protocols in the higher layers evolved to a much greater degree. As the transport layer welcomes QUIC[43], the application layer has accumulated an enormous amount of innovation and progress, in particular, onion routing and Tor[37, 22], Bitcoin[62], VPN protocols[65, 25] and decentralized instant chat[58, 81], that fight for digital rights and/or promote decentralization. Yet one must not overlook the crypto-constructs that lay the foundation for all of them, which I discuss in detail in Section 1.4.

Unfortunately again, because all of these are built on the Internet, a state-level censor could easily abuse its local authority to target them. A few regimes are notorious to have blocked a vast amount of them to various extents, employing complicated passive analysis and active probing techniques [27, 48, 85, 89, 29, 28]. Although a state would need to consider the collatoral damage, a totalitarian regime would not hesitate to block an entire protocol before it can figure out how to block it selectively [95, 89]. Apart from state-level actors, because commerical local ISPs additionally have an incentive in income and profit, not only do they perform surveillance and censorship like state-actors [10, 9], they also implement unjust policies easily with their local control of the infrastructure, like unfairly limiting the use of certain P2P protocols [24, 66, 60, 10]. Although ISPs argue that these P2P protocols can consume too much bandwidth, the other side of the story is that ISPs often oversubscribe and fraudulently advertise the bandwidth of Internet service they provide [69, 13]. This essentially is a probabilistic exploit on its customers — when almost all of the users happen to use the maximal bandwidth the ISP sells to them, congestion and throttling occur, and the users, not the ISP, ultimately pay the price — this is also the same kind of injustice imposed by airlines who oversell tickets.

The question of how to build protocols and systems that preserve one's rights on top of the Internet that powerful adversaries control, fighting against surveillance, censorship, and overall other unfair practices which the infrastructures of the current networks happen to enable, is central to my interest in the high-level design of secure distributed systems.

1.2 Data-driven analysis: Lights through artifical clouds

Despite my intention to understand systems, many of them are not open for analysis, for various reasons. In particular, the aforementioned systems for digital surveillance and censorship are not only proprietary but often state secrets. However, just as no material can perfectly insulate against heat in physics, no system can perfectly isolate information; some of it inevitably leaks through side channels (see Section 1.1.1).

By carefully observing their behaviors and probing their responses under controlled conditions, we can gain insight. Combined with data science techniques, these observations reveal hidden structures and enable us to draw meaningful conclusions.

The process usually involves 1) raising questions and claims about some properties of a target system 2) conducting experiments (often involving purchased servers within such as Aliyun) 3) analyzing data 4) answering questions and verifying claims. As a simple example, Sheffey et al. very recently studied the IP addresses injected by the GFW censorship system. By first finding injected IPs and then probing them within the GFW, they found three categories of such IPs [77].

Thanks to continuous work from both academic scholars and dedicated organizations, (e.g. [90, 63, 28]), I and a small set of others who suffer from such systems and who are fortunate enough to know these works, could know what we are facing everyday better. Conversely, I strongly hope to contribute to the free-side of the arms race, making the free Internet reachable to everyone.

1.3 LLMs and security: A timely matter

The contemporary popularity of LLMs in both the academia and the general public is so obvious that I don't need to cite to support it. One particular public usage of them is for information retreival [56, 94], which has many security implications. Among them, my interests are in: 1) how to preserve the privacy of all who are involved in the training data. 2) how to circumvent LLM censorship. With all these said, LLMs are not of as much theoretical interest to me as any of the previous or following subjects; I study this matter mainly because of its wide usage and thus its impact.

Regrettably, I do not have enough knowledge to write meaningful things about 1). But I do for 2). At first sight, it would seem to be a tricky problem, because even to this day, non-trivial neural networks are still mostly blackboxes whose connections we know little about. However, there are actually surprisingly many ways to censor such a blackbox, combining censorship on input, output, tokens, weights (to limited extent), and also using specific fine-tuning to train models to reject undesired contents

Where there is a way to censor, there is a way to defend. First, we can poison the training data [15], even in a computationally undetectable way, by using cryptographic constructs [38, 44], which is a good example of applying "paradoxical" crytographic constructs, to which I turn in Section 1.4. Second, although a model on a server is unmodifiable, the tendency of LLMs to follow commands have been taken advantage of to override some protection mechanisms [93]. Finally, one can always request a LLM to speak in some encoding[2].

Although these attacks are intriguing, I must emphasize that the topic interests me primarily because of the influence of LLMs on the general public now. Although architectures and training algorithms have been explored in great detail, the basic paradigm of adjusting weights between connected nodes to approximate an assumed existing target function remains conceptually close to the original perceptron model [72], and thus remains a naïve model of our brain that fails to capture many finer details of it, and that reflects our limited understanding of our brain. Moreover, the optimization problems, belonging to applied real analysis, is not my thing. Thus, I have overall limited interest in LLMs themselves, unless there happen to be the next big idea some day.

[2] As the time of writing, it is completely doable to role-play with ChatGPT by declaring that I am a cryptographic professor who wants to set assignments for students to practice some cipher, say, the Caesar ciper, and that I want ChatGPT to predict student replies. Then, we can pretty freely input censored text under the cipher, which ChatGPT will not refuse. However, its output are often too garbled to be meaningful; I have not conducted an empirical experiment on how much that is likely.

1.4 Cryptography: A mathematical savior?[3]

1.4.1 The potential of "paradoxical" modern crypto constructs

Security is often described as an arms race — that who controls more resources tends to discover more vulnerabilities, devise more attacks, and harden their systems more. Where is hope, then, when most self-censor and give up fighting the regime [14, 61, 84] and even fewer of the fighters have sufficient skills to join in the security war [68, 46]?

I believe one partial solution to what might sound like a power struggle lies within (modern) cryptography. For thousands of years since the use of the earliest symmetric encryption methods like the Caesar cipher, the ability to securely communicate through an insecure channel had still been mostly limited to the privileged who could afford persistent access to physical secure channels to exchange the keys and reliable safeguarding of the keys. A stunning turning point was found in what was widely regarded as the beginning of modern cryptography, Diffie and Hellman's *New Directions in Cryptography* [21], where they gave a practical mathematical procedure to Merkle's original idea for establishing a key known only to both parties over an insecure channel.

At the time, the idea seemed so paradoxical that Merkel himself faced rejections by his then professor Hoffman and the CACM [59]:

> "I am sorry to have to inform you that the paper is not in the main stream of present cryptography thinking and I would not recommend that it be publioned in the Communications of the ACM."
>
> "Experience shows that it is extremely dangerous to transmit key information in the clear."[59]

Even Shannon shared that assumption: "*The key must be transmitted by non-interceptible means from transmitting to receiving points*" [75, p. 670].

Such "paradoxical" ideas of modern cryptography are what attract me most to it, because the more paradoxical such an idea is, the more power it returns individuals to preserve their rights against overwhelmingly strong adversaries[4]. Thus, such ideas become the enabler that underlies those systems mentioned in section 1.1.2: TOR, Bitcoin, Matrix, etc., and the catalyst that leads to the massive adoption of society advancements such as e-commerce. Remarkably, in less than 50 years since 1976, we already have a number of such discoveries in addition, and here's a list of some of them.

Pseudorandom functions The kind of deterministic algorithms whose outputs look like that of a random oracle, by block ciphers like AES[18] or other ways [35], has a strong link with various other essential constructs like one-way functions.

Zero-knowledge proofs By interactively leveraging challenges that only a true prover could easily solve, zero-knowledge proof [39] enables one to

[3] I use the question mark, because cryptography cannot be, at least not a savior alone, if other problems are not solved. See the discussion at the end of this subsection.

[4] Albeit unfortunately they all rely on some assumptions, to which I return in Section 1.4.2.

verify that the prover knows a witness of a problem in NP [36] without revealing the witness.

Homomorphic encryptions Doing computation on encrypted data has been a very desirable property for many years. Although many earliest public-key encryption schemes, such as RSA [71] and ElGamal [26], already natively supported limited homomorphism like modular multiplication by their design, the first full scheme that allowed arbitrary computation on encrypted data was only proposed in 2009 [33].

These discoveries have many applications and implications, two of which I pay most attention to. Theoretically, they update our understanding of certain theoretical lower bounds of how much security one can achieve in the face of strong adversaries, no matter the power difference between them (provide the adversary is still bounded within feasible computational limits, of course). Socially, they help minimize the trust required to perform global-level of collaboration that involves people from vastly different backgrounds and beliefs; such a collaboration that would typically require a mutually trusted central authority to organize, now only requires the participants to trust a few foundamental assumptions of cryptography.

Personally, I would go a step further. I believe modern cryptography offers not only technical tools, but a possible mitigation for one of the big challenges our democracies face today: fragmentation, polarization, and the erosion of shared trust [30, 52, 64]. By design, cryptography redistributes power to a certain degree — governments need resources to crack one's system, be it burning a vulnerability, social engineering, or coercing, when they can't simply overcome the crypto barrier that unites people's power together, in the sense that massive invasion of digital rights requires the governments to multiply the efforts requires on a single person[5]. In this way, cryptography does something more profound, uniting people's power passively, when prospects of active collaboration are damaged by trust issues. Fortunately, as many modern infrastructures already deploy encryption and other crypto schemes by default, this "passive solidarity" is not a dream but a reachable reality, one that shows how mathematics can serve as a partial safeguard for democracy in an increasingly divided world.

1.4.2 Limitations and users' responsibilities

Having all hopes on passive constructs to save people's active lives is unrealistic. As a direct problem, crypto constructs all have assumptions, particularly that some problems are hard. Although P vs. NP seems rather remote to one's normal life, more serious limitations emerge when applying cryptography: that is when a theoretical model fails to capture all the danger that can break the assumptions. As a very practical example, I have been devoted to using OpenPGP

[5] This simplifies the issue by ignoring vulnerabilities, phishing techniques, etc., that can target many at once, but these are usually dealt with quickly once burned.

since I know it. One central assumption of it is that one's private key is never compromised. This rather simple assumption on paper brings a lot more serious problems when actually in use.

1. A OpenPGP secret is usually used for at least a few years. Although the risk of a system compromise at one time is low, it accumulates as time passes. Even as someone aware of security issues and practices, I have no confidence in that there will be no breach (more strictly speaking, opportunities of breaches) to my system once a few years.

2. As such, people have used many practices to safeguard the secret, such as writing it down physically and storing it in a vault, or use a more convenient media such as a hardware key. But if we want a key to be useful at all, we need to use it, and each use increases the risk of compromise; even if a hardware key prevents the secret from being extracted, a compromised system can still trick the key into performing operations, combined with some authentication breach.

3. An arguably safer way that reduces interaction with the secret is to have a master secret stored in isolation such as on a hardware key, and then create subkeys that are signed by the master secret and used daily. The master secret is only used to manage subkeys.

4. I personally use the above scheme, but a subkey's compromise can still be fatal in many cases. One specific problem is the lack of forward secrecy when a subkey is used for encryption. What's worse, one who only wants to snoop on my communication can stay low-profile, and I may never get to know a subkey is compromised before its expiration date.

Thus, although those "paradoxical" properties of crypto constructs are strong and many remain sound for many years, real world details really affect, and in many cases, decide how secure one's systems are in daily usage.

Since such issues are faced even by security researchers everyday, the general public clearly will suffer more from them. Anderson reported that cryptosystems in retail banking failed mostly because of "implementation errors and management failures," instead of cryptoanalysis problems, in 1993 [4]; and that observation remained mostly the same, 25 years later [49]. From another domain, Adams and Sasse surveyed users of password systems and argued that the then such systems lacked a user-centric design, also identifying a "vicious circle" where security policy makers and users lack a mutual understanding [3].

One direction to address this problem is to develop more easy-to-use, or in another phrase, more fool-proof systems that hide complexity from users [16]. often by combining crypto constructs in more complicated ways to patch known big issues. For instance, the forward secrecy problem is patched by most instant-messaging systems with real-time session key (ephemeral key) establishments. In particular, there is a recent trend to make systems "secure by default" [74]. That raises the important question: can security can really be achieved automatically and passively by default?

I argue that it cannot — there are essential complexities of security that should not be hidden. The use of ephemeral keys decouples encryption and decryption from the master secret, thereby protecting it from exposure through routine operations; but what stops a careless user from handing out freely what we try so hard to protect? And doesn't building more fool-proof systems risk spoiling users and decreasing the overall awareness of the responsibility that comes naturally when rights-preserving systems return power to users [31, 19]? Thus, I contend that we should instead build systems that encourages and directs users to actively engage with their own security. Consistent with my view, in the afore-cited work of Adams and Sasse, they did not advocate for providing for the users at all cost; instead, they suggested that many users are aware of security but need to be informed of the circumstances and be trained. Similarly, both Anderson and Køien, separated by 25 years, arrived at the same conclusion that there is a deficiency in security competence and it is imperative to educate and spread awareness more about security [4, 3]. Finally, I find the underlying reasoning very similar to that of why freedom is not free, to which I return in Section 2.3.

To end this section, cryptography is the enabler of and essential for the hope of the digitally abused to break free from the powerful adversaries, but on the other hand, it is not a savior for those who lack basic understanding of security. To apply cryptography in an engaging way that automatically incites users to learn more about security, is a central research direction I want to explore. Besides, because of its deep connection with complexity theory, number theory, and other theoretical fields, it also serves as a good preparation before entering these realms in the later stages of my academic career.

2 Software engineering: My earliest and continuing passion

2.1 Software construction

The seed for my passion in questioning how things should be built and be amazed at how complex things interact with each other when assembled together, was there in me long before I had an idea of what research or the academia were. In high school, I was obsessed with creating my own video games, what I hoped would be my sanctuary from my then miserable life. My unsuspectingly ambitious goal of starting from building a game engine unsurprisingly failed, but it did introduce me to the book *Game Engine Architecture* [40], which isn't academically significant, but its contents made me realize that I was more into how things work in the engine such as how threads coordinate with each other than into actually making a game.

This interest of figuring out how a complex system work and how to build such one, leads me into learning various software engineering principles, in particular, those of Liskov [54, 55], learning them deeper when offered by my undergraduate courses, and applying some of them in all of my big projects, since

I got to know them.

With all these principles, engineering of large software systems still remains a big challenge, particularly in interface design, separation of components, and all that we still have no fixed rules to follow. In other words, large systems engineering largely remains an art instead of a science [12, 76]. Could it become a science one day, governed by well-establioned rules that guide us toward optimal software systems in all or at least most situations? That is an intriguing question I aspire to explore while continuing to construct software in the following years.

2.2 Formal methods/verification

Many software engineering principles are concerned with mistakes made by humans. Not only do we try to minimize bugs from programs, we also need to reduce them from the specification or requirements of the software. Therefore, formalizing the specification and verifying that the software built indeed satisfies it could be seen as an ultimate way to eliminate bugs [42].

2.2.1 The speed problem of formalization

Perhaps unsurprisingly, the industry has a low usage of fully verified software [88], since the business dynamics changes rapidly, and approaches that allow quick iteration of software, such as agile development, are much more preferred [23]. However, even in the academia, formal verification is not frequent, outside of critical areas. One central reason could be that formalization of software is simply too time-consuming [88]; as a price of having reduced software logic into primitive steps, much more of such steps are required.

That might seem to be an essential difficulty preventing the wide-adoption of formal methods in software engineering, but another way of dealing with the problem is increasing the formalization speed. One approach is to formally build more advanced and verified steps out of primitive steps and define them in *proof assistants*. For example, Coq (Rocq) allows one to build custom tactics and use a meta-language such as Ltac to manipulate tactics at a higher level [8]. Furthermore, LLMs, the current hot topic, have the capacity to output seemingly correct formalization at high speed [67]. Combined with proof assistants, which can automatically verify the correctness of the output, we could have the potential to greatly speed up formalization of not only software, but also of informal mathematical proofs found everywhere in mathematical texts, if the product of the correct rate and the output speed of LLMs could surpass that of ours.

2.2.2 How formalization may change collaboration

One interesting application of proof assistants is Massot's `Blueprint` [57]. It is mostly a dependent graph that links each piece (lemma) of a big proof as its nodes, which one can click on to go to its rendered LaTeX, LaTeX source, and Lean (a proof assistant) source. Depending on a node's formalization status, it's displayed differently in the graph [79]. Beside being used in various math projects (see its README), it also has the potential to support building interactive math textbooks for students.

In another direction, the deep trust that was previously required for each collaborator's math skills, can now be reduced by automatic verification with proof assistants. Thus, whereas a research-level math project used to be done by a few professional mathematicians only, such one now can be instead conducted this way: 1) the few expert lead mathematicians design a whole picture of the theorem to prove and divide it into pieces 2) people, even those who are not professional mathematicians, can claim pieces that they think they can solve. 3) when they give solutions, it's the proof assistants' job to verify them. As a proof of concept, Tao has led a pilot project conducted such way [80].

While most software engineering projects require far less trust than a math project does, projects of engineering critical secure systems could still demand a high-level of trust and thus benefit from proof assistants in the same way as math projects above. In particular, could the development of a fully-verified and quite complete (in the sense of containing drivers, file systems, and other components, unlike a microkernel) OS kernel be possible, if collaborated this way? That would be an interesting question to answer in my future research.

2.3 Free software, free society[6]

As our lives are more digitalized, digital computers are playing an increasingly important role in them. Consequently, one who controls the computers will have great power over our lives. But do we control the computers (in the general sense, including mobile phones, smart devices, etc.) that we purchased and own? By the very definition, software controls our computers. Then, the question becomes, do we control the software that runs on our computers?

Unfortunately, the answer is no for proprietary software. Stallman has a number of nice essays on this topic [78], and I would just use a simple analogy here: using proprietary software is like entrusting one's money to another person, whose operations are opaque, whose decisions one has little or no influence on, whose interests are mainly profit, and who otherwise has no connection with one. Clearly, no sane person would choose to do that. The core problem of proprietary software is that via them the developers impose such an unjust power imbalance over its users, which tends to corrupt and lead to mistreatment of users, as history has repeatedly shown us.

By launching the free software movement, Stallman sought to restore power to users by securing their essential freedoms [78, Essay 1]. The movement has achieved remarkable successes, yet today it still faces serious obstacles. An prominent example is the current situation of open source, perhaps surprisingly. Often associated with Raymond's *The Cathedral and the Bazaar* [70], open source in fact emerged from within the free software community as a deviation

[6]The title here is the book title of the essays collection of Dr. Richard Stallman [78].

from its core philosophy. As Stallman emphasizes, "The term 'open source' quickly became associated with ideas and arguments based only on practical values, such as making or having powerful, reliable software" [78, Essay 14]. By shifting focus from freedom to utility, open source has obscured the very principles the free software movement was founded to protect.

There are two significant problems of the current situation of open source. The first is about the licenses. Some licenses only open the source code but prevents unauthorized modification and sharing; some others grant total access but fail to prevent the work from being stolen (i.e. creating a proprietary fork from it).

The second problem concerns the question whether freedom is passive and free. Many users of free or open source software almost never inspect, modify, or share the code [45], thereby remaining as dependent on developers as users of proprietary systems. Their situation differs only in that the developers of free software may act more ethically and the part of the users who do exercise their freedoms can somewhat prevent the developers from going rogue. Yet the open source movement's deliberate avoidance of freedom as a guiding value undermines awareness of it within the broader community. As much as convenience is important, one should realize that the underlying free software principles which encourages modification and sharing are the enabler of the Bazaar model. The danger of discarding those principles is that it creates an opportunity for corporate corruption and take over, undermining not only freedom, but eventually the convenience the users seek at the beginning. Today, corporate-led open-source projects, such as Microsoft's VS Code, illustrate this dynamic. Although they are sustained by community contributions, their governance, licensing, and branding remain firmly under corporate control [20, 17]. The result is software that strengthens corporate power while exploiting users (e.g. VSCode has built-in telemetry that is difficult to disable).

That situation of open source, in my observation, is dangerously similar to the rise of authoritarian populism discussed by Norris and Inglehard, who understand populism as "a style of rhetoric reflecting first-order principles about who should rule, claiming that legitimate power rests with 'the people' not the elites. It remains silent about second-order principles, concerning what should be done, what policies should be followed, what decisions should be made" [64, p. 4]. Similarly, I view most popular open source projects as reflecting shallow principles of allowing everyone to contribute to add more features, but ignoring deeper principles as to what direction the software should take, which decisions to make when convenience conflicts with freedom, and what are not allowed, even if they add more resources to the development of the software. Although a difference is that many populists reject the previous "ruling elites" whereas many open source members may not strongly oppose free software, they all involve people who commit to good principles such as democracy and open contribution only in the form, but ignoring the ideas and actions behind. Consequently, just as open source projects are corrupted and taken over by corporations, Norris and Inglehard observed that such shallow commitment to democracy in turn damages democracy itself and welcomes authoritarian [64].

Before concluding the discussion, one should note that there are many pieces of open source software which are very close to free software, which, although not under a strict free software license, grants most or in some cases, all freedoms to its users, and whose community do have people that care for these freedoms. For these pieces of software, does how we call them matter; and can we prefer the term "open source" over "free software" for popularity, since free software is always open? Stallman points out that it matters, "*because different words convey different ideas. While a free program by any other name would give you the same freedom today, establishing freedom in a lasting way depends above all on teaching people to value freedom*" [78, Essay 14].

Recall I argued in Section 1.4 that we should review our current trend of pursuing security by default, hiding all complexities in secure systems, and I suggested to engage users in security more. I believe this is essentially the same issue. Just as some previously free software opens the door to corruption by using the name "open source" that implicitly teaches people to prioritize convenience over freedom, the security community's focus to build more fool-proof systems sets users more remote from the what security is really about in its core, freedom from harm, power over one's life, and the responsibility and complexity that inherently come with it.

I see this as a profound yet subtle undermining of the free software movement, in which the core values of free software are silently lost amid the popular chase for convenience. Stallman quitted his scentist career at MIT and became an activist to dedicate his life to the free software movement. In the meantime, can we contribute to it from the academia? Could we find ways to build software that somehow helps users overcome the current trend of chasing popularity and make the importance of freedom as well as security appear pressing to them? And if we could, how can we ensure that users have incentives to use our software instead of the mainstream ones, even when they are aligned with the exploitative structures? These are not just important questions for my future research in free and secure software to answer, they are also essential for the society, if it is our goal as scientists to hope our knowledge will end up serving people, instead of becoming the tools that enslave them.

3 Mathematics: My purer philosophical pursuits

As much as I am driven by real-life problems to understand systems better, the force behind my sometimes unusally deep dive into even more primitive principles is my strong desire to understand the foundational reasonings. In this section, I will describe how that has led me to explore the purer mathematical areas and my current interests in them.

3.1 Type theory and metamathematics

In the manuals of various proof assistant languages, technical terms from type theory such as universe, sort, and introduction & elimination rules, are frequently mentioned. For me who is driven to understand what's going on behind, contact with type theory is my destiny. Furthermore, I ventured deeper into its history and the philosophical doctrines it was born from.

I have thought of a few ways to present this section, and in the end I decided that I will present it as a narrative of the basic history. If you share my curiosity for foundational reasonings, then after seeing the history I presented in my style, I believe you will also see my passion. A good starting point is probably Cantor's work on his theory of abstract sets, particularly his ingenious abstraction of sets to cardinal numbers for their comparsion. In 1638 Galileo noted that there is a bijection between the natural numbers and their squares: $f : n \mapsto n^2$, raising questions about the principle that the whole is greater than any of its parts. Cantor took this further to systematically compare sets, using bijections as the primary tool.

An essential mental construct that Cantor used was his diagnoal argument, which is often presented when proving the real numbers is not countable. Yet the central philosophical reasoning behind this argument applies to general sets which we don't know are countable or not, and this leads to this important theorem, whose proof I present because it's short and raises questions about many things.

Cantor's Theorem. *There is no surjection from any set S to the set of all of its subsets $\mathcal{P}(S)$. In Cantor's words, the cardinal number of $\mathcal{P}(S)$ is always larger than that of S.*

Proof. When S is empty, it is trival that there's no surjection between $\emptyset$ and $\{\emptyset\}$. Then we consider when S is not empty.

Suppose for now there exists a bijection $f : S \to \mathcal{P}(S)$. f's being a bijection enables us to apply Cantor's diagonal argument, because now we can make sure the number of rows and columns are the same: arrange such a matrix that each column corresponds to an element $s \in S$, and each row is arranged to correspond to the sequence $f(s)$ for all s, in the same order as the columns. Along the diagonal, flip each element to define this subset $T \subset S$ such that $s \in T$ iff $s \notin f(s)$. Although we used the bijection fact to get to this definition, the definition itself does not require a f's being a bijection; it can be any function $f : S \to \mathcal{P}(S)$. But we will need it to be a surjection for the argument that follows. So we now demote f to be any surjection.

Because f is a surjection, there is such s_0 that $f(s_0) = T$. Now we ask if $s_0 \in T$. Note that here we implicitly assume either $s_0 \in T$ or $s_0 \notin T$. In either case, by definition of T, we will arrive at a contradiction. Thus, by *reductio ad absurdum*, it is not the case that there exists such a surjection f. $\square$

Cantor and Russell arrived at two similar paradoxes, both involving the set of all sets.

Cantor's paradox Consider the set of all sets Ω and its power set $\mathcal{P}(\Omega)$. By Cantor's theorem, the latter should be larger than the former. On the other hand, $\mathcal{P}(\Omega)$, being a set of sets, should be a subset of Ω, a contradiction.

Russell's paradox When Russell was later analyzing Cantor's theorem, he wondered what if S itself was Ω. Under the assumption that all objects are sets, we thus have $S = \mathcal{P}(S)$, and the simple $f : s \mapsto s$ serves as a bijection between them. In that case, Cantor's diagnoal subset T becomes the famous form that we know of: $T := \{s \mid s \notin s\}$.

For Russell, his paradox was discovered at a bad time, when he was on his way of finishing his *Principles of Mathematics*, and when he assumed that there was a universal set. He had hoped for a single, unified solution to the paradoxes of logic. But to his discouragement, after a few years of various attempts to fix them in the way he hoped, he was eventually forced to come to his ramified theory of types, using a hierachy of types to classify propositions [82].

Perhaps one reason why the paradoxes were so difficult to fix was that Cantor's diagonal method had revealed something deeply wrong about the previous mathematical reasoning, one which goes deeper than even the prominent work of axiomatic set theory which tried to eliminate these paradoxes. Previously, mathematicians thought that they were working with real and complete math objects such as sets and numbers and that they were speaking truths of them as theorems. Since that had lead to contradictions, there must have been something wrong in the foundation of what we believed these objects to be or our logic. Even if we discard the previous foundation entirely and adopt the axiomatic theory as the new foundation, there must still be the matter of truth and meaning — we cannot adopt the axiomatic set theory entirely formally as the new foundation, without a meaning about what we think is true or not [47, Ch. 12].

Unfinioned.

Part II

Additional information

4 My interest in teaching and assistantship

Although I have not had any experience as a true teaching assistant, I have always been attracted to the prospect of teaching others with my knowledge. In fact, when I am learning something exciting, I will often imagine presenting it to others and be excited about it if I believe I found a nice way to do that. (You know, I believe Section 1.1.1 and 1.4 are such examples. Why not take a look if you have not already?) Similarly, when I am writing something that I want others to see, like this document, I often take the position of the reader and wonder if she will be touched by my writing and if she can get my ideas clearly.

To talk more specifically, I am particularly fond of Prof. Winston's teaching style, which I discovered when I started self-learning his 6.034 at MIT [86]. One important thing I learned from him is that he demonstrated passion consistently during his teaching, whereas some other professors I experienced have chosen to present things more objectively. That comparison made a huge difference in my perspective — I feel more engaged and inspired by his teaching style. I also went on to listen to his *How to Speak* lecture [87] after finishing learning his course.

5 My career plan

I plan to enter the academia after my PhD and become a professor somewhere there is enough freedom of academic expression. The condition is important because I come from a place where politics and ideology alignment is the priority, and I see firsthand how damaging that is.

My plan to become a professor is a result of these three things:

1. My strong academic intersets and my passion in teaching as described early.

2. My wish to spread knowledge, advancing my own as well as the others' understanding of the subjects I study.

3. Finally, my personal commitment to live and work ethically in a way that does not require freedom to come at someone else's expense. It may be easier to gain freedom by aligning with exploitative structures, but I cannot accept that path. I hope to take a different approach: to find and strengthen spaces within academia where knowledge can be shared openly, collaboration can be fair, and students can pursue ideas without fear or compromise. Though I recognize that academia is not perfect and can be exploitative (to a less degree), I believe it still offers the best opportunity

to practice and promote a more just environment in which I work and devote my passion.

References

[1] Abelson, H., Anderson, R., Bellovin, S. M., Benaloh, J., Blaze, M., Callas, J., Diffie, W., Landau, S., Neumann, P. G., Rivest, R. L., Schiller, J. I., Schneier, B., Teague, V., and Troncoso, C. Bugs in our pockets: the risks of client-side scanning. *Journal of Cybersecurity 10*, 1 (01 2024), tyad020.

[2] Abelson, H., Anderson, R., Bellovin, S. M., Benaloh, J., Blaze, M., Diffie, W., Gilmore, J., Green, M., Landau, S., Neumann, P. G., Rivest, R. L., Schiller, J. I., Schneier, B., Specter, M. A., and Weitzner, D. J. Keys under doormats: mandating insecurity by requiring government access to all data and communications ‡. *Journal of Cybersecurity 1*, 1 (11 2015), 69–79.

[3] Adams, A., and Sasse, M. A. Users are not the enemy. *Commun. ACM 42*, 12 (Dec. 1999), 40–46.

[4] Anderson, R. Why cryptosystems fail. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (New York, NY, USA, 1993), CCS '93, Association for Computing Machinery, p. 215–227.

[5] Anderson, R. Chat control or child protection? 2210.08958, 2022. https://arxiv.org/abs/2210.08958.

[6] Anderson, R. Privacy and freedom issues. https://www.cl.cam.ac.uk/archive/rja14/#Lib, 2024. Accessed 7 Oct 2025.

[7] Anonymous. The internet coup. Technical analysis, InterSecLab, sep 2025. https://interseclab.org/research/the-internet-coup/. Accessed 25 Sept 2025.

[8] Barras, B., Boutin, S., Cornes, C., Courant, J., Coscoy, Y., Delahaye, D., de Rauglaudre, D., Filliâtre, J.-C., Giménez, E., Herbelin, H., et al. The coq proof assistant reference manual. *INRIA, version 6*, 11 (1999), 17–21.

[9] Becker, E., and Djuitcheu, H. Could your mobile broadband internet provider threaten your digital privacy? In *2022 Workshop on Next Generation Networks and Applications (NGNA 2022)* (2022).

[10] Bendrath, R., and Mueller, M. The end of the net as we know it? deep packet inspection and internet governance. *New Media & Society 13*, 7 (2011), 1142–1160.

[11] Boneh, D., DeMillo, R., and Lipton, R. On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology 14* (mar 2000), 101–119.

[12] Brooks, F. P., and Bullet, N. S. Essence and accidents of software engineering. *IEEE computer 20*, 4 (1987), 10–19.

[13] Cachon, G. P., and Feldman, P. Pricing services subject to congestion: Charge per-use fees or sell subscriptions? *Manufacturing & Service Operations Management 13*, 2 (2011), 244–260.

[14] Chen, X., Xie, J., Wang, Z., Shen, B., and Zhou, Z. How we express ourselves freely: Censorship, self-censorship, and anti-censorship on a chinese social media. In *Information for a Better World: Normality, Virtuality, Physicality, Inclusivity* (Cham, 2023), I. Sserwanga, A. Goulding, H. Moulaison-Sandy, J. T. Du, A. L. Soares, V. Hessami, and R. D. Frank, Eds., Springer Nature Switzerland, pp. 93–108.

[15] Cinà, A. E., Grosse, K., Demontis, A., Vascon, S., Zellinger, W., Moser, B. A., Oprea, A., Biggio, B., Pelillo, M., and Roli, F. Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Comput. Surv. 55*, 13s (July 2023).

[16] Cranor, L. F. *Security and usability: designing secure systems that people can use.* " O'Reilly Media, Inc.", 2005.

[17] CTOL Editors. Beware of fake open-source: How hidden proprietary api traps are undermining trust in software. `https://www.ctol.digital/news/beware-fake-open-source-hidden-proprietary-traps-undermining-trust-software/`, aug 2024. Accessed 2 Oct 2025.

[18] Daemen, J., and Rijmen, V. Aes proposal: Rijndael. Gaithersburg, MD, USA, 1999.

[19] Di Nocera, F., and Tempestini, G. Getting rid of the usability/security trade-off: A behavioral approach. *Journal of Cybersecurity and Privacy 2*, 2 (2022), 245–256.

[20] Dias, L. F., Steinmacher, I., and Pinto, G. Who drives company-owned oss projects: internal or external members? *Journal of the Brazilian Computer Society 24* (dec 2018).

[21] Diffie, W., and Hellman, M. E. *New Directions in Cryptography*, 1 ed. Association for Computing Machinery, New York, NY, USA, 2022, p. 365–390.

[22] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The Second-Generation onion router. In *13th USENIX Security Symposium (USENIX Security 04)* (San Diego, CA, Aug. 2004), USENIX Association.

[23] Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N. B., et al. A decade of agile methodologies: Towards explaining agile software development. *Journal of systems and software 85*, 6 (2012), 1213.

[24] Dischinger, M., Mislove, A., Haeberlen, A., and Gummadi, K. P. Detecting bittorrent blocking. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement* (New York, NY, USA, 2008), IMC '08, Association for Computing Machinery, p. 3–8.

[25] Donenfeld, J. A. Wireguard: Next generation kernel network tunnel. In *NDSS* (2017), pp. 1–12.

[26] Elgamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory 31*, 4 (1985), 469–472.

[27] Elmenhorst, K., Schütz, B., Aschenbruck, N., and Basso, S. Web censorship measurements of http/3 over quic. In *Proceedings of the 21st ACM Internet Measurement Conference* (New York, NY, USA, 2021), IMC '21, Association for Computing Machinery, p. 276–282.

[28] Ensafi, R., Fifield, D., Winter, P., Feamster, N., Weaver, N., and Paxson, V. Examining how the great firewall discovers hidden circumvention servers. In *Proceedings of the 2015 Internet Measurement Conference* (New York, NY, USA, 2015), IMC '15, Association for Computing Machinery, p. 445–458.

[29] Ensafi, R., Winter, P., Mueen, A., and Crandall, J. R. Analyzing the great firewall of china over space and time. In *Proceedings on Privacy Enhancing Technologies* (2015), vol. 2015, pp. 61–76.

[30] Fukuyama, F. *Trust: The social virtues and the creation of prosperity.* Simon and Schuster, 1996.

[31] Furnell, S. Usable cybersecurity: a contradiction in terms? *Interacting with Computers 36*, 1 (01 2024), 3–15.

[32] Genkin, D., Shamir, A., and Tromer, E. Rsa key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology – CRYPTO 2014* (Berlin, Heidelberg, 2014), J. A. Garay and R. Gennaro, Eds., Springer Berlin Heidelberg, pp. 444–461.

[33] Gentry, C. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 2009), STOC '09, Association for Computing Machinery, p. 169–178.

[34] Głowacka, D., Youngs, R., Pintea, A., and Wołosik, E. Digital technologies as a means of repression and social control. *Policy Department for External Relations, Directorate General for External Policies of the Union 1* (2021), 1–106.

[35] Goldreich, O., Goldwasser, S., and Micali, S. How to construct random functions. *J. ACM 33*, 4 (Aug. 1986), 792–807.

[36] Goldreich, O., Micali, S., and Wigderson, A. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM 38*, 3 (July 1991), 690–728.

[37] Goldschlag, D., Reed, M., and Syverson, P. Onion routing. *Commun. ACM 42*, 2 (Feb. 1999), 39–41.

[38] Goldwasser, S., Kim, M. P., Vaikuntanathan, V., and Zamir, O. Planting undetectable backdoors in machine learning models : [extended abstract]. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (2022), pp. 931–942.

[39] Goldwasser, S., Micali, S., and Rackoff, C. *The knowledge complexity of interactive proof-systems*. Association for Computing Machinery, New York, NY, USA, 2019, p. 203–225.

[40] Gregory, J. *Game engine architecture*. AK Peters/CRC Press, 2018.

[41] Handley, M. Why the internet only just works. *BT Technology Journal 24*, 3 (2006), 119–129.

[42] Hoare, C. A. R. An axiomatic basis for computer programming. *Commun. ACM 12*, 10 (Oct. 1969), 576–580.

[43] Iyengar, J., and Thomson, M. *RFC 9000 QUIC: A UDP-Based Multiplexed and Secure Transport*. Internet Engineering Task Force (IETF), may 2021. `https://www.rfc-editor.org/rfc/rfc9000.html`.

[44] Kalavasis, A., Karbasi, A., Oikonomou, A., Sotiraki, K., Velegkas, G., and Zampetakis, M. Injecting undetectable backdoors in obfuscated neural networks and language models. In *Advances in Neural Information Processing Systems* (2024), A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37, Curran Associates, Inc., pp. 21537–21571.

[45] Kaur, R., Kaur Chahal, K., and Saini, M. Understanding community participation and engagement in open source software projects: A systematic mapping study. *Journal of King Saud University - Computer and Information Sciences 34*, 7 (2022), 4607–4625.

[46] Kazansky, B. Digital security in context: Learning how human rights defenders. *Tactical Technology Collective* (2016). `https://cdn.ttc.io/s/secresearch.tacticaltech.org/pages/pdfs/original/DigitalSecurityInContext.pdf`.

[47] Kleene, S. C. *Introduction to Metamathematics*. P. Noordhoff N.V., Groningen, 1974.

[48] Knockel, J., Crandall, J. R., and Saia, J. Three researchers, five conjectures: An empirical analysis of {TOM-Skype} censorship and surveillance. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI 11)* (2011).

[49] Kø ien, G. M. Why cryptosystems fail revisited. *Wireless Personal Communications 106* (may 2015), 85–117.

[50] Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., and Yarom, Y. Spectre attacks: exploiting speculative execution. *Commun. ACM 63*, 7 (June 2020), 93–101.

[51] Kocher, P., Jaffe, J., and Jun, B. Differential power analysis. In *Advances in Cryptology — CRYPTO' 99* (Berlin, Heidelberg, 1999), M. Wiener, Ed., Springer Berlin Heidelberg, pp. 388–397.

[52] Levitsky, S., and Ziblatt, D. *How democracies die.* Crown, 2019.

[53] Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Horn, J., Mangard, S., Kocher, P., Genkin, D., Yarom, Y., Hamburg, M., and Strackx, R. Meltdown: reading kernel memory from user space. *Commun. ACM 63*, 6 (May 2020), 46–56.

[54] Liskov, B., and Zilles, S. Programming with abstract data types. *SIGPLAN Not. 9*, 4 (Mar. 1974), 50–59.

[55] Liskov, B. H., and Wing, J. M. A behavioral notion of subtyping. *ACM Trans. Program. Lang. Syst. 16*, 6 (Nov. 1994), 1811–1841.

[56] Liu, Z., Zhou, Y., Zhu, Y., Lian, J., Li, C., Dou, Z., Lian, D., and Nie, J.-Y. Information retrieval meets large language models. In *Companion Proceedings of the ACM Web Conference 2024* (New York, NY, USA, 2024), WWW '24, Association for Computing Machinery, p. 1586–1589.

[57] Massot, P. Lean blueprints. `https://github.com/PatrickMassot/leanblueprint?tab=readme-ov-file`, 2025. Accessed 4 Oct 2025.

[58] Matrix.org. Matrix for instant messaging. `https://matrix.org/docs/chat_basics/matrix-for-im/`, 2023. Accessed 26 Sept 2025.

[59] Merkle, R. C. Publishing a new idea. `https://ralphmerkle.com/1974/`. Accessed 25 Sept 2025.

[60] Mondal, A., Trestian, I., Qin, Z., and Kuzmanovic, A. P2p as a cdn: A new service model for file sharing. *Computer Networks 56*, 14 (2012), 3233–3246.

[61] Moore-Gilbert, K., and Abdul-Nabi, Z. Authoritarian downgrading, (self)censorship and new media activism after the arab spring. *New Media & Society 23*, 5 (2021), 875–893.

[62] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Available at SSRN 3440802* (2008).

[63] Niaki, A. A., Hoang, N. P., Gill, P., Houmansadr, A., et al. Triplet censors: Demystifying great {Firewall's}{DNS} censorship behavior. In *10th USENIX workshop on free and open communications on the internet (FOCI 20)* (2020).

[64] Norris, P., and Inglehart, R. *Cultural backlash: Trump, Brexit, and authoritarian populism.* Cambridge University Press, 2019.

[65] OpenVPN, Inc. Openvpn. `https://openvpn.net/`, 2025. Accessed 26 Sept 2025.

[66] Piatek, M., Madhyastha, H. V., John, J. P., Krishnamurthy, A., and Anderson, T. E. Pitfalls for isp-friendly p2p design. In *HotNets* (2009).

[67] Polu, S., and Sutskever, I. Generative language modeling for automated theorem proving, 2020.

[68] Rahman, Z., Thompson, N., Walker, T., and Kaminski-Killany, K. Technology tools in human rights. Tech report, The Engine Room, 2016. `www.theengineroom.org/wp-content/uploads/2017/01/technology-tools-in-human-rights_high-quality.pdf`.

[69] Raju, A., Gonçalves, V., Lindmark, S., and Ballon, P. Evaluating impacts of oversubscription on future internet business models. In *NETWORKING 2012 Workshops* (Berlin, Heidelberg, 2012), Z. Becvar, R. Bestak, and L. Kencl, Eds., Springer Berlin Heidelberg, pp. 105–112.

[70] Raymond, E. The cathedral and the bazaar. *Knowledge, Technology & Policy 12* (sep 1999), 23–49.

[71] Rivest, R. L., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM 21*, 2 (Feb. 1978), 120–126.

[72] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review 65* (1958), 386–408.

[73] Rosson, Z., Anthonio, F., and Tackett, C. Emboldened offenders, endangered communities: internet shutdowns in 2024. Technical report, AccessNow, feb 2025. `https://www.accessnow.org/press-release/keepiton-internet-shutdowns-2024-en/`. Accessed 25 Sept 2025.

[74] Ruohonen, J. Sok: The design paradigm of safe and secure defaults. *Journal of Information Security and Applications 90* (2025), 103989.

[75] Shannon, C. E. Communication theory of secrecy systems. *The Bell System Technical Journal 28*, 4 (1949), 656–715.

[76] Shaw, M. Prospects for an engineering discipline of software. *IEEE Software 7*, 6 (1990), 15–24.

[77] Sheffey, J., Zohaib, A., Kang, D., Durumeric, Z., Houmansadr, A., and Wu, Q. Extended abstract: I'll shake your hand: What happens after DNS poisoning. In *Free and Open Communications on the Internet* (2025).

[78] Stallman, R. M. *Free software, free society: Selected essays of Richard M. Stallman*. GNU, 2002.

[79] Tao, T. Formalizing the proof of pfr in lean4 using blueprint: a short tour. `https://terrytao.wordpress.com/2023/11/18/formalizing-the-proof-of-pfr-in-lean4-using-blueprint-a-short-tour/`, nov 2023. Accessed 4 Oct 2025.

[80] Tao, T. A pilot project in universal algebra to explore new ways to collaborate and use machine assistance? `https://terrytao.wordpress.com/2024/09/25/a-pilot-project-in-universal-algebra-to-explore-new-ways-to-collaborate-and-use-machine-assistance/`, sep 2024. Accessed 4 Oct 2025.

[81] TokTok. A new kind of instant messaging. `https://tox.chat/`, 2025. Accessed 26 Sept 2025.

[82] Urquhart, A. *The Cambridge Companion to Bertrand Russell*. Cambridge Companions to Philosophy. Cambridge University Press, 2003, ch. 8, pp. 286–310.

[83] Wagner, B., Bronowicka, J., Berger, C., Behrndt, T., et al. Surveillance and censorship: The impact of technologies on human rights. EESC: European Economic and Social Committee, 2015.

[84] Wang, M., and Mayer, J. Self-censorship under law: A case study of the hong kong national security law, 2023.

[85] Wendzel, S., Volpert, S., Zillien, S., Lenz, J., Rünz, P., and Caviglione, L. A survey of internet censorship and its measurement: Methodology, trends, and challenges, 2025.

[86] Winston, P. H. 6.034 artificial intelligence. `https://ocw.mit.edu/courses/6-034-artificial-intelligence-fall-2010/`, 2010. Accessed 4 Oct 2025.

[87] Winston, P. H. Res.tll-005 how to speak. `https://ocw.mit.edu/courses/res-tll-005-how-to-speak-january-iap-2018/`, 2018. Accessed 4 Oct 2025.

[88] Woodcock, J., Larsen, P. G., Bicarregui, J., and Fitzgerald, J. Formal methods: Practice and experience. *ACM Comput. Surv. 41*, 4 (Oct. 2009).

[89] Wu, M., Sippe, J., Sivakumar, D., Burg, J., Anderson, P., Wang, X., Bock, K., Houmansadr, A., Levin, D., and Wustrow, E. How the great firewall of china detects and blocks fully encrypted traffic. In *32nd USENIX Security Symposium (USENIX Security 23)* (Anaheim, CA, Aug. 2023), USENIX Association, pp. 2653–2670.

[90] Wu, M., Zohaib, A., Durumeric, Z., Houmansadr, A., and Wustrow, E. A wall behind a wall: Emerging regional censorship in china. In *2025 IEEE Symposium on Security and Privacy (SP)* (2025), pp. 1363–1380.

[91] Xue, D., Ablove, A., Ramesh, R., Danciu, G. K., and Ensafi, R. Bridging barriers: A survey of challenges and priorities in the censorship circumvention landscape. In *33rd USENIX Security Symposium (USENIX Security 24)* (Philadelphia, PA, Aug. 2024), USENIX Association, pp. 2671–2688.

[92] Yarom, Y., and Falkner, K. FLUSH+RELOAD: A high resolution, low noise, l3 cache Side-Channel attack. In *23rd USENIX Security Symposium (USENIX Security 14)* (San Diego, CA, Aug. 2014), USENIX Association, pp. 719–732.

[93] Yi, S., Liu, Y., Sun, Z., Cong, T., He, X., Song, J., Xu, K., and Li, Q. Jailbreak attacks and defenses against large language models: A survey, 2024.

[94] Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Chen, H., Liu, Z., Dou, Z., and Wen, J.-R. Large language models for information retrieval: A survey. *ACM Transactions on Information Systems* (Sept. 2025).

[95] Zohaib, A., Zao, Q., Sippe, J., Alaraj, A., Houmansadr, A., Durumeric, Z., and Wustrow, E. Exposing and circumventing {SNI-based}{QUIC} censorship of the great firewall of china. In *34th USENIX Security Symposium (USENIX Security 25)* (2025), pp. 783–802.