Contents

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 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's rights are increasingly threatened by the powerful digitally [24, 61, 54]. A prominent example is the continuous push for exceptional access to to people's data and communication in various forms by governments[1], ignoring expert opinions against such measures [2, 1, 3]. I unfortunately has suffered from more aggressive attacks, primarily systematic, comprehensive, and far-reaching censorship [4] [67, Sect. 5], because of my background.

[1] For 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 A high-level view of the Internet

A high-level understanding of how the digital infrastructures work is essential to understand what enables these threats. 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 in its address 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 [30], 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.*" [30]. On the other hand, the protocols in the higher layers evolved to a much greater degree. As the transport layer welcomes QUIC[32], the application layer has accumulated an enormous amount of innovation and progress, in particular, onion routing and Tor[27, 14], Bitcoin[45], VPN protocols[47, 17] and decentralized instant chat[41, 59], 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 1.3.

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 [19, 35, 63, 65, 21, 20], 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 [69, 65]. 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 [7, 6], they also implement unjust policies easily with their local control of the infrastructure, like unfairly limiting the use of certain P2P protocols [16, 48, 43, 7]. 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 [51, 10]. 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, cogestion 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 preserves 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 secure distributed systems.

1.1.2 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 she wants her 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 [38]. In fact, Meltdown touches more than logical flaws in hardware; it relies on the timing difference of cache accesses to extract information from it [68], one that directly connects with the physical world. At this point, I feel that I should also mention the Spectre attacks family [36], which, unlike Meltdown, does not exploit a clear logic bug or fault in the hardware. Instead, they 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 [37, 8, 22, 68], 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.

1.2 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. On the other hand, similar to side channel attacks, these systems inevitably leak information in different ways.

By carefully observing their behavior 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 [57].

Thanks to continuous work from both academic scholars and dedicated organizations, (e.g. [66, 46, 20]), 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 Cryptography: A mathematical savior

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 [11, 44, 62] and even fewer of the fighters have sufficient skills to join in the crypto war [50, 33]?

I believe one 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 Ceasar 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, Deffie and Hellman's *New Directions in Cryptography* [13], 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 first glance, the idea sounded so impossible that Merkel himself faced rejections when presenting the idea to his then professor Hoffman and to the CACM [42]:

> "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 published in the Communications of the ACM."
>
> "Experience shows that it is extremely dangerous to transmit key information in the clear."[42]

The rejections represented a concensus of the old cryptography community that even Shannon concurred with: "*The key must be transmitted by non-interceptible means from transmitting to receiving points*" [55, p. 670], which demonstrates how "paradoxical" the idea was.

Such "paradoxical" ideas of modern cryptography are what attract me most to it, because they are the enabler that underlies those systems mentioned in section 1.1: TOR, Bitcoin, OpenVPN, 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[12] or other ways [25], 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 [28] enables one to verify that the prover knows a witness of a problem in NP [26] 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 [53] and ElGamal [18], 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 [23].

These discoveries have many applications and implications, two of which I pay most attention to. One is that 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. The other is how 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 strong central authority to organize, now only requires the participate 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 solution for addressing one of the big challenges our democracies face today: fragmentation, polarization, and the erosion of shared trust. By design, cryptography redistributes power — it makes mass surveillance prohibitively costly, even for state-level actors, by forcing them to target individuals rather than populations. For example, suppose breaking one person's encryption, on average, required a single day of dedicated effort (through side-channels, vulnerabilities, or social engineering rather than mathematics), that constraint alone would prevent massive surveillance, since no adversary has that many days or resources to do this on everyone. In this way, encryption does something profound: it automatically and passively unites individuals, protecting each not by active coordination that is increaingly difficult, but by the collective shield of widespread adoption. 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 safeguard for democracy in an increasingly divided world.

To end this section, cryptography not only flows towards my passion for purer philosophical understanding, because of its deep connection with theoretical computer science and mathematics, to which I turn in Section 3, but also carries my hope to break free from the strong grasp of the tyrannies today, one that represents my free will which refuses to lose my agency and independence, in a world that too often seeks to reduce us into interchangable screws for the system.

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. I was obsessed with creating my own video games, what I hoped would be my sanctuary from my 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* [29], 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 [39, 40], 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 [9, 56]. Could it become a science one day, governed by general principles that guide us toward optimal software systems? 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, formalize the specification and verify that the software built indeed satisfies it could be seen as an ultimate way to eliminate bugs [31].

Perhaps unsurprisingly, the industry has a low usage of fully verified software [64], since the business dynamics changes rapidly, and approaches that allow quick iteration of software, such as agile development, are much more preferred [15]. However, even in the academia, formal verification is not frequent, outside of critical areas. One central reason could be that formalization of software is too time consuming [64]; as a price of having reduced software logic into primitive steps, 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 way 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 [5]. Furthermore, LLMs, the current hot topic, have the capacity to output seemingly correct formalization at high speed [49]. Combined with proof assistants, which can automatically verify the correctness of the output, we could have the potential to greatly speed up not only formalization of software, but that 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.

2.3 Free software, free society[2]

As our lives are more digitalized, digital computers are playing an increasingly important role in them. Consequently, she 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 [58], and I would just use a simple analogy here: using proprietary software is like entrusting one's money to another person, whose operations are obaque, 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 starting the free software movement, Stallman tried to give the power back to users by respecting the users' essential freedoms [58, Essay 1]. His movement has had remarkable achievements, but there are deep issues about the society that free software alone cannot solve. The concept that is most popular today is open source software, the form that most attribute to the *Cathedral and the Bazaar* essay [52], but which, according to Stallman, actually was a group of the original free software community that deviated from the core philosophy [58, Essay 14].

Unfinished.

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.

[2]The title here is the book title of the essays collection of Dr. Richard Stallman [58].

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 [60].

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. To avoid the paradoxes, one prominent work was Zermelo's axiomatic set theory (later refined by others) which restricted what sets can be. But there are still solved problems. 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 [34, Ch. 12].

Unfinished.

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] ANDERSON, R. Chat control or child protection? 2210.08958, 2022. `https://arxiv.org/abs/2210.08958`.

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

[5] 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.

[6] 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).

[7] 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.

[8] BONEH, D., DEMILLO, R., AND LIPTON, R. On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology 14* (mar 2000), 101–119.

[9] BROOKS, F. P., AND BULLET, N. S. Essence and accidents of software engineering. *IEEE computer 20*, 4 (1987), 10–19.

[10] 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.

[11] 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.

[12] DAEMEN, J., AND RIJMEN, V. Aes proposal: Rijndael. Gaithersburg, MD, USA, 1999.

[13] DIFFIE, W., AND HELLMAN, M. E. *New Directions in Cryptography*, 1 ed. Association for Computing Machinery, New York, NY, USA, 2022, p. 365–390.

[14] 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.

[15] 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.

[16] 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.

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

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

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

[28] 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.

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

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

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

[32] 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`.

[33] 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`.

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

[35] 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).

[36] 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.

[37] 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.

[38] 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.

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

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

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

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

[43] 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.

[44] 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.

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

[46] 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).

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

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

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

[50] 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`.

[51] 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.

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

[53] 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.

[54] 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.

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

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

[57] 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).

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

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

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

[61] 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.

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

[63] 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.

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

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.