A system capable of verifiably and privately screening global DNA synthesis

Carsten Baum[1,2], Jens Berlips[3], Walther Chen[3], Hongrui Cui[4], Ivan Damgard[1,*], Jiangbin Dong[5], Kevin M. Esvelt[3,6,*], Mingyu Gao[5,12], Dana Gretton[3,6], Leonard Foner[3], Martin Kysel[3], Kaiyi Zhang[4], Juanru Li[4], Xiang Li[5], Omer Paneth[7], Ronald L. Rivest[7], Francesca Sage-Ling[3], Adi Shamir[8], Yue Shen[10], Meicen Sun[11], Vinod Vaikuntanathan[7], Lynn Van Hauwe[3], Theia Vogel[3], Benjamin Weinstein-Raun[3], Yun Wang[10], Daniel Wichs[9], Stephen Wooster[3], Andrew C. Yao[3,5,12,*], Yu Yu[4,12], and Haoling Zhang[10]

[1]Department of Computer Science, Aarhus University, Denmark
[2]DTU Compute, Technical University of Denmark, Denmark
[3]SecureDNA Foundation, Switzerland
[4]Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[5]Institute for Interdisciplinary Information Sciences, Tsinghua University, China
[6]Media Lab, Massachusetts Institute of Technology, USA
[7]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA
[8]Department of Applied Mathematics, Weizmann Institute of Science, Israel
[9]Department of Computer Science, Northeastern University, USA
[10]China National GeneBank, China
[11]Department of Political Science, Massachusetts Institute of Technology, USA
[12]Shanghai Qi Zhi Institute, China
*ivan@cs.au.dk
*andrewcyao@mail.tsinghua.edu.cn
*esvelt@mit.edu

Summary

A free DNA screening system based on multi-party oblivious hashing preserves customer privacy while verifiably checking gene and oligonucleotide synthesis orders at high speed with a negligible false alarm rate.

Abstract

Printing custom DNA sequences is essential to scientific and biomedical research, but the technology can be used to manufacture plagues as well as cures. Just as ink printers recognize and reject attempts to counterfeit money, DNA synthesizers and assemblers should deny unauthorized requests to make viral DNA that could be used to ignite a pandemic. There are three complications. First, we don't need to quickly update printers to deal with newly discovered currencies, whereas we regularly learn of new viruses and other biological threats. Second, anti-counterfeiting specifications on a local printer can't be extracted and misused by malicious actors, unlike information on biological threats. Finally, any screening must keep the inspected DNA sequences private, as they may constitute valuable trade secrets. Here we describe SecureDNA, a free, privacy-preserving, and fully automated system capable of verifiably screening all DNA synthesis orders of 30+ base pairs against an up-to-date database of hazards, and its operational performance and specificity when applied to 67 million base pairs of DNA synthesized by providers in the United States, Europe, and China.

Introduction

Custom DNA synthesis is foundational to biomedical research, underpinning everything from cancer immunotherapies to SARS-CoV-2 vaccines. However, the same technology can also be used to produce pathogens[1–3]. The first infectious virus to be assembled from synthetic DNA was generated in 2002 at a cost of more than $10 per base pair[4]. Just over two decades later, the price has fallen more than a thousandfold, the number of individuals with the necessary skills has grown from dozens to many thousands[5], and a pandemic has directly and indirectly killed over 20 million people[6].

Numerous well-intentioned research programs aim to identify viruses capable of causing new pandemics and share their genome sequences[7–11]. One already maintains a public list of viruses ranked by threat level[12]. Logically, superior understanding and continual improvements in biological programming will eventually allow the generation of engineered pandemic-class pathogens.

The obvious solution to the resulting proliferation problem, screening all DNA synthesis for hazards, was first recommended in 2006[13]. Remarkably, the members of the International Gene Synthesis Consortium (IGSC) voluntarily monitor an estimated 80% of global DNA synthesis, even though the relative cost of human-assisted screening is rising with volume[14,15].

Nevertheless, synthesis screening remains an unsolved problem that threatens biotechnology and the world:

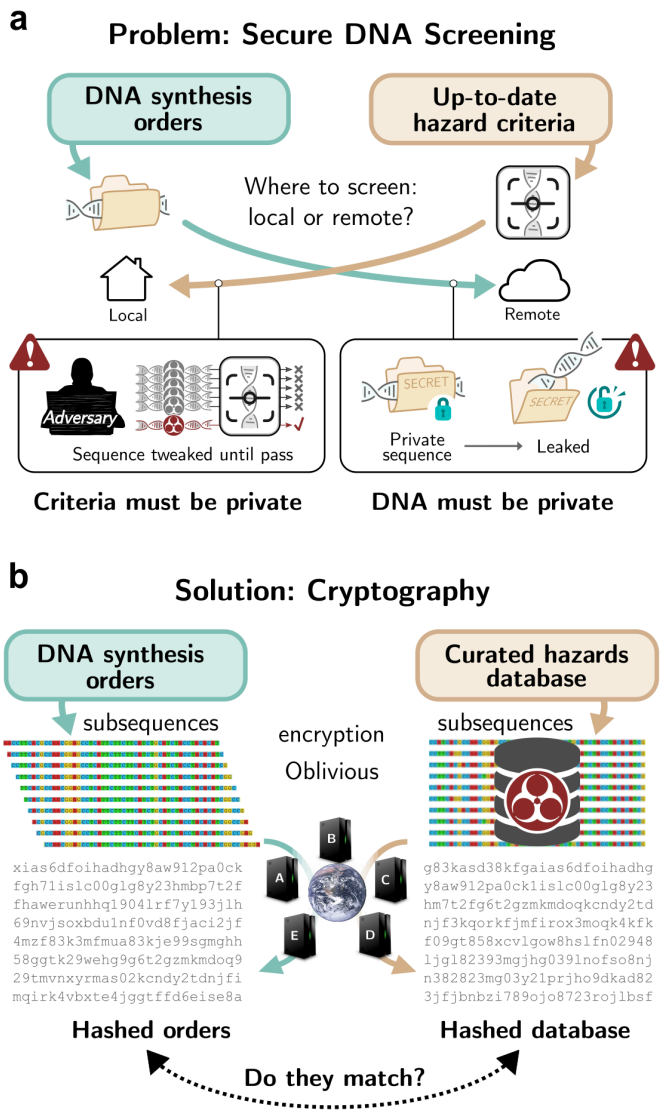


Fig. 1 | | Securing DNA synthesis screening.
a) Secure and universal DNA synthesis screening requires a way to verifiably determine whether DNA synthesis orders correspond to hazardous biological functions without disclosing anything else about the orders or what defines a hazard. Disclosing a private order may compromise trade secrets, while leaking the criteria would make it possible to evade screening. In the companion paper, we describe how to convert this challenge into an exact-match computer science problem by pre-defining wild-type sequences and predicted variants that exhibit hazardous functions.
b) The SecureDNA system allows synthesizers and database contributors to *obliviously* perform one-way transformations of their subsequence windows, which can be directly compared to find any matches. The database provides the synthesizer with a timestamped verification that *n* windows sent by the synthesizer were screened against a particular database version.

- Chatbots suggest ordering from non-members[16]
- Short sequences pose hazards but are not screened[14]
- There is no way to check if screening is up-to-date
- Firms may be liable because they cannot verify that they performed best-in-practice screening
- Customers value benchtop synths to protect trade secrets, but benchtops cannot be reliably screened[17]

A reliable DNA synthesis screening system must address three key design challenges:

1. *Bio-design*: translate the biological problem of hazard recognition into a computer science problem
2. *Crypto-design:* devise a way to screen that protects the privacy of orders and of the database
3. *System-design*: implement an automated system capable of verifiably and securely screening all DNA synthesis worldwide

In a companion paper, we describe a candidate solution to the bio-design challenge[18]. Random Adversarial Threshold (RAT) screening is an algorithmic approach that identifies exact matches to essential subsequences of hazards as well as pre-computed functional variants of those subsequences, excluding those present in harmless genes. Because RAT screening reliably detects hazards without flagging innocuous sequences, it may not require expert humans to check its work and can in principle be automated, including on benchtop devices.

Here we address the cryptographic and systems design challenges (Fig. 1) using a novel application of oblivious cryptography that enables private hazard screening (Fig. 2). Our complete implementation, including a graphical user interface, screens at high speed and low cost while demonstrating very high specificity (Fig. 3), and implements a certificate system permitting authorized laboratories to access hazardous DNA (Fig. 4).

Results

Theoretical crypto-design and analysis

Suppose that a Customer places an order for sequence *s* with a Synthesizer, who wants to know whether it's safe to make *s* according to an up-to-date remote Database *D* of known harmful DNA and peptide subsequences (Fig. 2a, Appendix A) while remaining secure against eavesdroppers. The Synthesizer can ask the Database to check whether any of the constant-length DNA and translated peptide subsequences found in *s* are present in *D* (Fig. 2b). The crypto-design challenge is to find a way for the Database to answer without 1) learning anything about *s* or 2) revealing anything about *D* beyond conveying the yes/no answer, even if some of the parties are compromised.

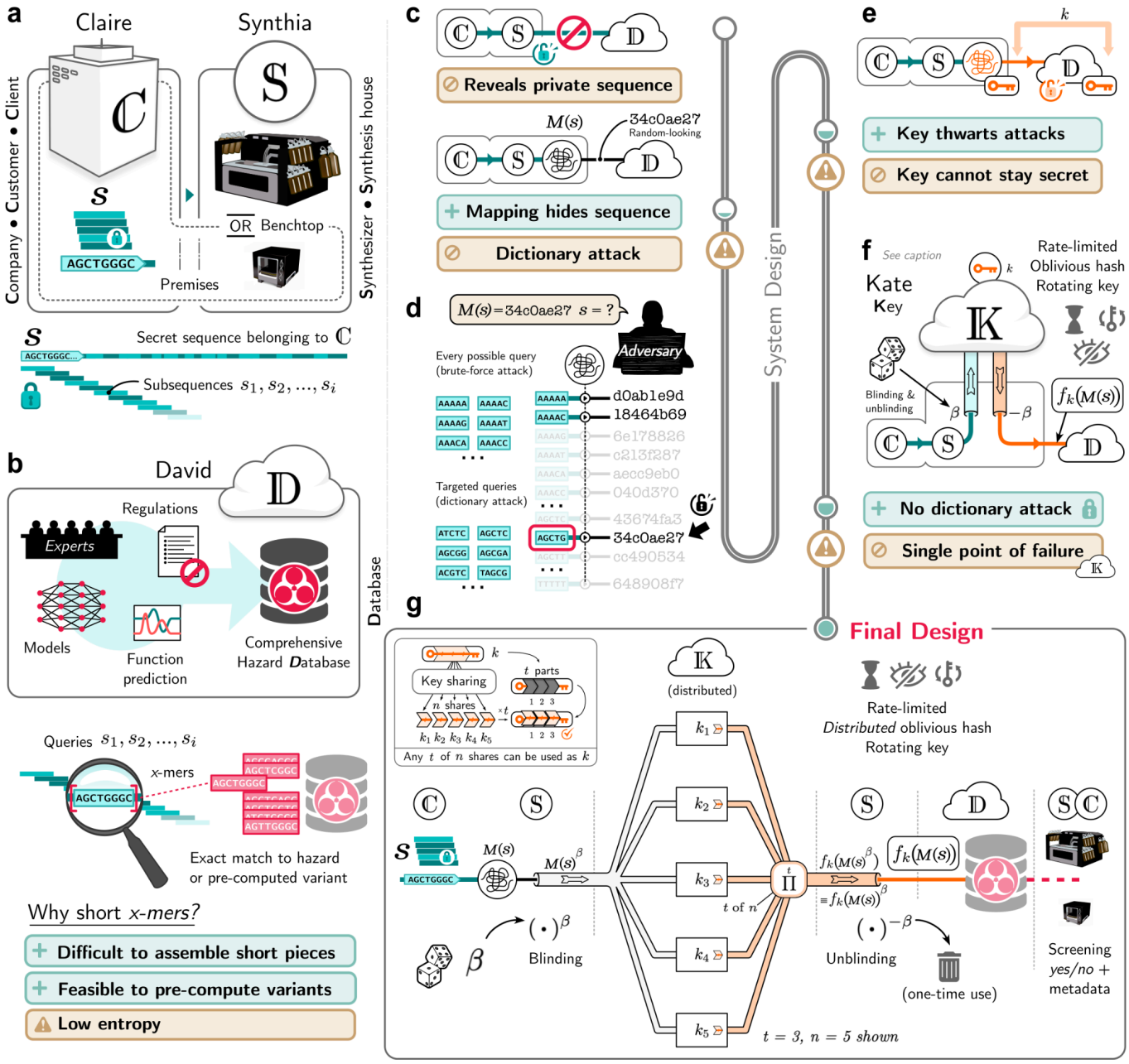


Fig. 2 Cryptographic challenges and solutions for secure DNA synthesis screening
a) Client Claire orders DNA from commercial synthesis house or benchtop Synthesizer Synthia. Claire's private DNA sequence *s* is split into all subsequences of length *x* (*x*-mers). **b)** In the simplest version, a centralized hazard Database David is populated with *x*-mers from hazards and predicted functional variants. Synthia sends *x*-mers to be screened, and David detects exact *x*-mer matches. If *x* is short, David can screen small oligonucleotides that might otherwise be assembled into hazards. **c)** Synthia cannot send *s* directly to David (as in current remote DNA screening systems) without violating Claire's privacy, but she could obscure *s* with a cryptographic mapping $M(s)$ and compare to David's similarly hashed sequences. **d)** A standard hash is easily cracked when the input is short or has low entropy using a brute-force or dictionary attack (trying many possible inputs). **e)** A keyed hash with a secret key *k* can thwart dictionary attacks, but requires that *k* be known to both Synthia and David, in which case Synthia could interrogate *D* and David could crack *s*. **f)** Keyserver Kate helps Synthia compute keyed hash $f_k(M(s))$ with oblivious cryptographic hashing. Synthia "blinds" the sequence *s* with a random value β before Kate applies *k*, then unblinds it afterward, so no eavesdropper (including Kate) can learn *s*. Kate can limit the rate of evaluation of $f_k(s)$, making dictionary attacks impossible. However, Kate is a single point of failure: an adversary compromising Kate gains *k*, and if Kate is offline, global synthesis stops. **g)** In the final design, $f_k(M(s))$ is distributed, making Kate's role robust. *k* is split into *n* key shares among *n* separated servers. Evaluating $f_k(M(s))$ requires a threshold number *t* of the *n* shares (n = 5, t = 3 shown). Up to *n*-*t* keyservers may be offline. *k* is rotated, e.g. biweekly. Any adversary must compromise *t* servers simultaneously between key rotations to steal *k*.

The simplest way for the Synthesizer and Database to determine whether any of their DNA or peptide subsequences are identical without learning anything else is to transform them using a one-way hash mapping and compare the hash outputs *M(s)* and *M(D)* (Fig. 2c).

But either of them could rapidly enumerate all possible subsequences, letting them decrypt *s* and *D* (Fig. 2d). This remains true if they use a hash based on key *k*, because both must know *k* (Fig. 2e). To restore privacy, we introduce the Keyserver, who uses a frequently-changed key *k* to help perform hashes – but only to the extent required to accommodate the maximum plausible rate of DNA synthesis within a given timeframe. This renders enumeration attacks infeasible (Fig. 2f). The Synthesizer and Keyserver can compute hash outputs *obliviously* using cryptographic techniques. Obliviousness means the Keyserver learns nothing about the subsequences in *s* or even the resulting hash outputs, while the Synthesizer learns only the hash outputs.

To allow the Database to identify matches, we give him a table *H* comprising the oblivious keyed hashes of all elements of the plaintext database *D*. He does not learn the content of *D*, only its hashes. Whenever the Synthesizer submits a hashed query, the Database can tell her if there are any matches to *H*. He learns nothing about her sequences and need not know any hazards.

Who, then, knows the plaintext list of hazards *D* and which subsequence windows are defended with many functional variants? By design, no one. Since we demonstrably cannot prevent pandemics from killing millions, it would be nice if we did not need to make emerging future threats credible by including them in a public screening list. This system would empower life scientists who spot new threats to update the database without either Keyserver or Database learning of them, and optionally without any of the contributing scientists learning of the additions made by anyone else.

Crucially, the Database can verify that screening was performed by sending the Synthesizer a time stamped signature attesting that a number of subsequences were screened against a particular version of *H* in compliance with local legal requirements. The Synthesizer can store this and relevant order information to demonstrate that best practices were followed, offering protection from liability in the event of misuse.

To summarize the baseline system:

- Scientists and Keyserver build and update Database *H* of obliviously hashed subsequences from hazards
- Customer orders sequence *s* from Synthesizer
- Synthesizer+Keyserver obliviously hash subsequences
- Synthesizer sends the hash outputs to Database
- Database tells Synthesizer if any are found in *H*
- If there are matches, Synthesizer refuses to make *s*

Security analysis of baseline screening

In the *semi-honest* model of cryptographic security, all parties follow the protocol, but may eavesdrop. The Keyserver learns nothing about *s* and *D* because the hash is oblivious. The Synthesizer learns whether *s* contains hazardous subsequences, but that is the point of screening. If no hazards are found, the Database learns nothing of *s* beyond its total length (which is biologically irrelevant), and possibly whether any parts of *s* are shared with hashed sequences from other clients. If *s* does contain hazards, he learns which subsequences match a specific *harmful subset* of subsequences from one (or more) hazards in *D*. He may also find statistical correlations between the harmful subsets in *H*, but remains ignorant of the plaintext hazards in *D*.

In the *malicious* model, a party may deviate from the protocol to obtain sensitive information or actively sabotage its execution. Note first the important premise that the Synthesizer wants to avoid creating hazards. Any synthesizer that is not hardware-locked can opt out of the protocol entirely and generate any hazardous DNA sequence they want, but this is clearly something no software can prevent. A Synthesizer that participates dishonestly can also disclose *s*, but the number and rate of queries they can make to *H* to reconstruct the content of *D* are limited. If the Database is corrupt, he can disclose *H* and any statistical correlations, but cannot interrogate *s* or *D* without help from the Keyserver.

But the Keyserver is different: if corrupt (either semi-honest or malicious), she can use *k* to hash all possible subsequences and thereby determine *s* and *D*.

Distributing the key

The integrity of the Keyserver is so vital that we safeguard her role by dividing the key *k* into *n* shares and distributing them across *n* keyservers using Shamir secret sharing[19] (Fig. 2g). The secret-sharing uses a threshold value *t,* such that any group of less than *t* keyservers never learn anything about *k*, while *t* or more keyservers can jointly perform an oblivious hash, playing the same role as the single keyserver did before. To

prevent a malicious party who gradually corrupts a majority of the keyservers from stealing *k*, the system coordinates periodic updates of the key shares using a process with the same security guarantees as the distributed oblivious hashing (i.e. no one ever reconstructs *k*). This achieves what cryptographers call *proactive security*: *k*, and thereby all *s* and *D*, remain protected even if all keyservers are eventually hacked as long as they are not compromised at the same time. The full system is described in detail in Appendix B.

System design and implementation

SecureDNA screens for genes and genomes corresponding to all Australia Group, ITAR, Chinese, and EU-listed export-controlled toxins and pathogens, all viruses described as potential pandemic pathogens in the scientific literature, all known viruses capable of human-to-human transmission, and all known viruses able to cause illness in humans. Each is tagged by region to ensure flags and denials comply with local regulations.

Hazardous sequences are partitioned into 30 or 42 bp DNA subsequences and 20 amino acid peptides. Mutants of 42-mers and predicted functional variants of peptides, which are computed based on protein structure and sequence homology, are generated for a subset. This process is performed in both directions of the original sequence[18]. When adapted for use with benchtop synthesizers, SecureDNA also screens permutations of bases to prevent users from swapping A's and C's in the reagent bottles and also in *s* (Extended Data Fig. 1).

To remove likely false alarms caused by flagging sequences common to hazards and non-hazards, each candidate subsequence is checked against a dataset of nucleotides and proteins derived from NCBI's nr/nt GenBank and GenPept databases, with the matched non-hazard subsequences removed from the hazards database. Non-hazard records are identified using a combination of taxonomic analysis (ignoring any matches within the same genus where appropriate), tell-tale keywords present in the dataset entry's description (e.g. "synthetic" or "recombinant"), and fraction of subsequences matching a hazard.

System operation

Each synthesizer or provider running the open-source *synthclient* program converts FASTA files from sequence orders into all eligible subsequences and permutations and translates them in all possible reading frames.

Next, *synthclient* contacts the keyservers to obliviously hash each subsequence (no DNA or translation is sent), then sends the hashes to the database server for comparison. Screening a typical-length query completes faster than most web pages load (Fig. 3a). The database server returns "accepted", "alert", or "denied", a decision which is final for benchtops and advisory for centralized providers. It also provides 1) a signed and timestamped verification that screening was performed using a specific database version, 2) information sufficient to produce a visual depiction of any submitted subsequences that exactly match any public hazard, and 3) notes relevant export control restrictions (Fig. 3b).

System performance

Despite modest resource provisioning (~$8K/yr, Methods), our alpha prototype is capable of screening 10^{10} base pairs annually at speeds faster than the typical DNA synthesis provider website loading time (Fig. 3a), providing customers with immediate feedback (Fig. 3b).

Synthclient runs at approximately 1100 bp/sec/CPU core on a machine supplied by the provider. The keyservers run at 4400 bp/sec (on very modest 4-core CPUs) and the DB runs at 41,500 bp/sec/disk (physical NVMe) on machines operated by SecureDNA. A very large provider screening $3*10^{11}$ base pairs per year would need three $2,000 16-core desktop-class CPUs or a cloud-based VPS costing $4,000/year. One such desktop would suffice for most providers. A benchtop synthesizer can utilize a $50 Raspberry Pi 4B (Extended Data Fig. 2).

A one-time investment of $50,000 for Keyserver and Database hardware will enable the current implementation to accommodate several trillion base pairs per year, which is easily sufficient to screen all global DNA synthesis for the next five years.

Quantifying specificity using real-world synthesis data

To measure specificity in a real-world context, we screened anonymized orders for over 150,000 genes and oligonucleotides synthesized by providers in the United States, Europe, and China, which together comprise over 61 million base-pairs generated for biological research and 6 million for DNA storage.

Over 99% of sequences passed screening, and an overwhelming majority (96%) featured zero matches. Retaining curated entries in our test hazard database for analysis, we observed that among sequences with one or more matches (4%), two-thirds (2.6%) were curated and would never have been flagged in practice (Fig. 3c).

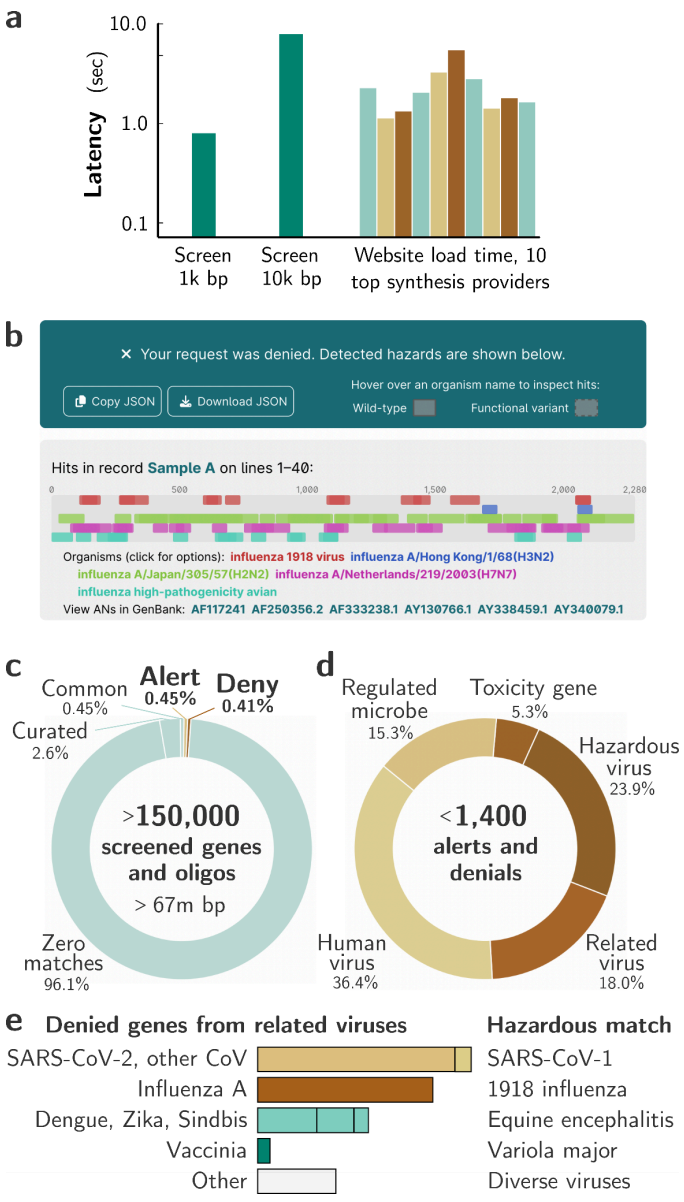


Fig. 3 | SecureDNA latency, output and quantified specificity on real-world synthetic genes.
a) A comparison of the time required to screen 1,000 and 10,000 base pair orders to the typical response times of randomly chosen DNA synthesis provider websites. SecureDNA does not meaningfully delay order placement and can return hazard assessments to customers in real-time, unlike approaches that rely on BLAST and expert human curation. **b)** SecureDNA returns wild-type matches and predicted functional variants of all windows matching public hazards. A hover-over interface can view the windows matching each hazard separately or go to the relevant GenBank file. This sequence was denied because it is a chimera of SARS-CoV-1 spike (middle portion) and a bat CoV.

One-third of those remaining (0.45%) were flagged as "Common" because they matched one of the many frequently used sequences in biotechnology that originated in a regulated organism or a human-infecting virus. Examples include the porcine teschovirus 2A peptide and the cytomegalovirus enhancer/promoter.

SecureDNA identified genes with at least one wild-type potential hazard or functional variant subsequence in 0.86% of sequences (Fig. 3c). Each was investigated by running BLAST on the entire gene sequence, then on any subsequences that were not clearly identified in the first attempt, and followed by manual labeling.

We would have alerted the provider but granted permission for 51.7% of non-"Common" genes with at least one non-curated match (Fig. 3d). 36% were from viruses capable of transmission in humans that are not currently regulated, while 15% matched genetic sequences from regulated microbes that could not confer toxicity to avirulent strains.

We would have denied 47% of genes with matches: 5% conferred toxicity upon an avirulent microbe, 24% most closely matched a regulated or potential pandemic human, animal, or plant virus, and 18% corresponded to closely related viruses (Fig. 3d). Because many of these appeared to be repeat orders in which a single customer ordered numerous variants of a gene, we estimate that fewer than 100 DNA synthesis orders worldwide would receive a first-pass denial on any given day (see below).

Crucially, we observed zero random false alarms: none of the DNA storage orders were flagged, and every denied gene was closely related to the hazard with which it shared windows. Of the related viruses that were denied, virtually all were pathogens capable of causing human or animal disease (Fig. 3e). Just as with regulated agents, laboratories working with these pathogens should already have permission from a biosafety authority. It is imperative for these authorized laboratories to obtain the DNA that they need to advance human knowledge and develop therapies without delay.

c) Analyzing >67 million base pairs of synthesized genes from multiple providers with SecureDNA flagged 0.91% (yellow) and denied 0.41% of genes (brown).
d) Investigations via iterative manual BLAST revealed that zero alerts or denials were true false alarms: all were from pathogens closely related to the matching hazard.
e) Related viruses, all capable of human infection, with sequences flagged for denial due to their similarity to still more hazardous relatives.

Automating customer screening and permissions

Members of the International Gene Synthesis Consortium spend considerable time and effort screening their customers for legitimacy. In some cases, providers contact the customer's local biosafety authority to ask whether the customer is authorized to work with a particular hazardous organism.

An ideal screening system would automatically grant researchers access to DNA corresponding to pathogens that their biosafety authority has already approved them to work with. SecureDNA allows researchers to submit an "exemption list token" (ELT) with their order to obtain any permitted hazardous DNA (Fig. 4a). A one-time token contains a public key identification (PKI) certificate instructing the database to ignore matches corresponding to exempted hazards, but only for requests originating from the user. This allows researchers to swiftly obtain DNA corresponding to any hazardous gene, organism, or set of raw sequences by requesting an exemption from their biosafety officer, who approves it using their own security key and a certificate issued by a higher biosafety authority, such as one at the national level (Fig. 4b, Extended Data Fig. 4).

To facilitate seamless ordering, we wrote software to help biosafety authorities convert approved laboratory research registration documents into exemption lists of permitted genes and organisms, then issue the lab an ELT (Fig. 4c). The principal investigator can separately grant certificates to lab members, enabling them to automatically bypass SecureDNA denials of sequences on the lab exemption list as long as they are shipped to the lab's address (Fig. 4c). In all cases, a physical hardware authentication key ($20-$30) is required to validate the user's identity, and placing an order for an exempted hazard notifies the principal investigator, the biosafety officer, and the organization's legal contact.

For example, suppose that a laboratory wants to develop a poultry vaccine and needs access to avian influenza, including high-pathogenicity strains. Using the exemption request tool, they can select "generic influenza A", which grants access to all known influenza A subsequences except those unique to hazardous strains, and also "high-pathogenicity avian influenza", then send the request to their biosafety authority (Extended Data Fig. 4). Once approved, every lab member can use the ELT and their personal certificate proving lab affiliation to ship any DNA sequences corresponding to those strains to the lab's registered address, or synthesize them on a benchtop device.

Researchers can also request exemptions covering lists of sequences, such as oligonucleotide libraries for deep mutational scanning[20] and directed evolution[21] (Fig. 4d).

As a final precaution, whenever an order for a public hazard is automatically approved using an exemption list, the SecureDNA database notifies the lab's principal investigator, biosafety authority, and institutional legal department for record-keeping purposes. Because biosafety vetting and approval are available as a commercial service[22], researchers anywhere in the world can access the benefits of seamless and secure ordering via automated customer screening.

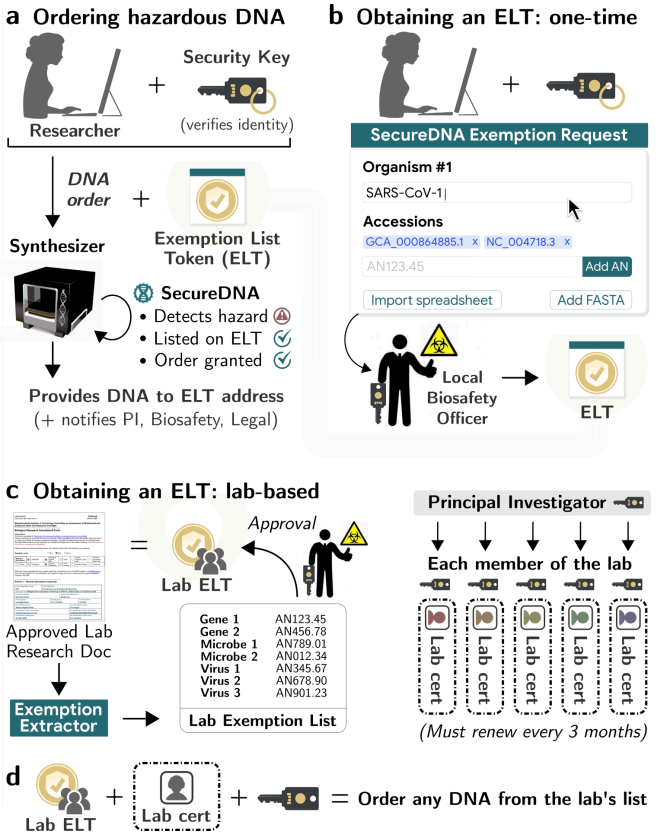


Fig. 4 | SecureDNA exemption lists for automated customer screening. a) Researchers with a hardware security key can submit an *exemption list token* (ELT) with their order to bypass denials for any hazard listed on the ELT. **b)** To obtain a one-time ELT, researchers can specify a hazard from a drop-down table, enter a GenBank accession number, or import a DNA sequence, then send it to their local biosafety officer to be approved and signed using the institution's certificate. **c)** Labs can obtain an ELT that will work for all members by using an extraction tool on their research registration document to obtain a list of genes, microbes, and viruses listed in the document, which can then be approved by the biosafety officer. The lab principal investigator can grant 'lab certificates' to each member. **d)** A lab ELT, lab cert, and matching security key can obtain DNA from any gene or organism the lab is approved to work with.

Discussion

Progress in the life sciences is demonstrably vulnerable to public mistrust[23]. A pandemic deliberately caused by a malicious actor would almost certainly trigger a backlash and draconian restrictions. Safeguarding the promise of biotechnology requires a way to limit access to legitimate laboratories[24,25], but the complexity of the problem and the expense of regulation have deterred international action.

The advent of free, automated, benchtop-compatible, and privacy-preserving DNA synthesis screening may allow nations currently hesitant to place their own companies at a competitive disadvantage to begin regulating the sector and clarify liability in the event of misuse (Extended Data Table 1). Indeed, signatories of the Biological Weapons Convention may be obligated to require free screening under Article IV, which states that parties must "take any necessary measures to prohibit and prevent the development, production, stockpiling, acquisition, or retention of the agents, toxins, weapons, equipment and means of delivery… within the territory of such State, under its jurisdiction or under its control anywhere."

While DNA synthesis screening may prevent widespread access to credible pandemic agents for many years, advances in *de novo* protein design[26–28] will gradually undermine its effectiveness. Design models can already generate functional equivalents of binding proteins; while this presumably includes toxins, they are far less relevant to international security than pandemics. However, *de novo* biodesign tools may eventually be capable of generating allosteric or catalytic proteins sufficient to enhance natural pathogens or even produce novel pandemic agents. Function and folding prediction tools that rely on a complete and intact sequence will not be able to detect such threats if the resulting synthetic genes are ordered in pieces, among other evasive strategies (Extended Data Fig. 5).

Controlling access to protein design tools via APIs and logging hazardous designs for screening purposes, which could be done even if source code was freely shared among developers, may help prevent designed hazards from evading detection. To further mitigate security risks, leaders in the protein design community have called for the retention of cryptographic records of DNA synthesis orders[29], which could deter malicious actors by reliably identifying the source of the harmful DNA after the fact. Storing hashes and distributing each set of rotated keys among trusted parties for release after a catastrophic event can permit past orders to be screened for future hazards while preserving privacy.

Collectively, our results suggest that SecureDNA can provide free, private, reliable, and verifiably up-to-date nucleic acid synthesis screening at a scale sufficient to meet global demand. If supported by favorable regulatory and liability policies, hardware integration into next-generation synthesizers, and subsidized trade-in programs, near-universal screening could dramatically reduce unauthorized access to pandemic-class agents without delaying research. By preventing the conversion of hazardous blueprints into dangerous pathogens, we can safeguard biotechnology and the world from the threat of deliberate pandemics.

Methods

Software

The SecureDNA code base is written in Rust, a "memory-safe" language designed to prevent a wide variety of typical programmer errors which account for roughly half of all security issues, such as mishandling storage allocation or exceeding array bounds[30]. All code, except that used for database generation, is available at https://github.com/SecureDNA/. The public demo is hosted at https://pages.securedna.org/staging/demo/.

We implemented the alpha prototype using Amazon Web Services (AWS) as specified:

- Synthesizer: one `c5d.2xlarge` instance with 8 CPU cores of 3.0 GHz and 16 GiB memory
- Database Server: one `c4.2xlarge` instance with 8 CPU cores of 2.9 GHz, 15 GiB memory, and gp3 disk
- Keyservers: three servers of type `t3.xlarge` with 4 CPU cores of 2.5 GHz and 16 GiB memory

To perform screening on a FASTA file, `synthclient` opens network streams to (at least) a threshold number of keyservers and a single database. It calls `quickdna` to generate subsequences for each window of the relevant size, each of which is obliviously hashed with the keyservers, unblinds the output, and sends the result to the database, which responds "Yes," "Yes, but" (EL), or "No". For a public hazard (currently all database entries), a denial is accompanied by index information, allowing the frontend of `synthclient` to present a visual analysis for the user.

Database generation

Listed U.S. Select Agent, Australia Group, EU, and Chinese pathogens were given region tags and separated based on whether they came from viruses, toxins, or microbes. All 30-mer, 42-mer, and 60mer windows from viruses, toxins, and genes capable of conferring toxicity to an avirulent microbe (e.g. the three toxin and five capsular genes of *B. anthracis*) were extracted and the 60mers translated into peptides. All single mutants of 42-mers were included along with a number of additional 30-mer and 42-mer predicted functional variants. Peptide windows were selected quasi-randomly and functional variants predicted using a combination of funtrp and BLOSUM62. For genes from regulated pathogens not linked to toxicity, one 42-mer for every 39-45 nucleotides was included and tagged "Regulated but Pass" to identify non-hazardous sequences from regulated organisms. Hazardous subsequences commonly used in biotechnology were tagged as "Common" and were not used to generate variants or trigger denials during screening. Finally, all subsequences with Shannon entropy below 1.6, often found in many unrelated organisms, were removed.

Database curation

Non-redundant nucleotide (nr/nt) and protein databases were downloaded from NCBI and subjected to taxonomic and keyword analysis to detect relatedness to hazardous genes and functions. The remaining putative harmless sequences were separated into 30-mer, 42-mer, and 20aa windows grouped by accession number (AN). Candidate subsequences from hazards were compared to the database of putative harmless sequences. Matches to putative harmless ANs that exceeded a threshold number of matches to a single candidate hazard were ignored as too closely related. For all remaining matches, the responsible subsequence was removed from the hazards database to minimize false alarms.

Performance

To determine server provisioning requirements, base pairs screened per second (bp/s) were quantified for each component by issuing randomized screening or cryptographic protocol requests as appropriate. A proprietary dataset of over 42,000 real customer orders was screened using the SecureDNA alpha prototype to assess performance under ideal conditions on realistic production traffic. To model benchtop synthesis conditions, wall-clock and user mode execution times were measured on a Raspberry Pi 4 client connected via WiFi screening random DNA sequences from 100 to 100,000 bp in length. The Raspberry Pi demonstrated performance exceeding 100 bp/s per core, indicating feasibility for low-cost embedded CPUs (Extended Data Fig. 2). Together these complementary tests quantified end-to-end latency, server capacity needs, and embedded processor utilization relevant to global deployment.

Sensitivity challenges

Three proprietary datasets generated by DNA synthesis providers were used for challenge testing. A positive control dataset including sequences from all species on the United States, EU, and Australia Group control lists was used to verify that all hazards were detected. A second test set included controlled and non-controlled sequences lightly manipulated for obfuscation. The final dataset included interspersed subsequences of controlled and non-controlled sequences to test sensitivity for

x-mers of different lengths. These tests were performed in addition to those described in the companion paper[18].

Specificity testing

Anonymized proprietary data from commercially synthesized genes were cryptographically screened via SecureDNA without disclosing confidential information. Results were conveyed to industry partners for analysis. Analyses of each individual gene flagged or denied by SecureDNA were conducted via nucleotide and/or protein BLAST, with as-yet-unidentified regions subjected to iterative analyses until all subsequences were identified. The SecureDNA results were then scored to quantify specificity.

Exemption lists

ELTs are cryptographically-signed objects containing the exemption list signed by the last certificate in a chain extending to the root held by the SecureDNA Foundation (Extended Data Fig. 3). Requests are generated using the browser-accessible `ELTR` tool, and can be approved by biosafety authorities and ELTs issued using the `elgui` tool.

Acknowledgements

We are deeply grateful to several anonymous industry partners, and to B. Mueller for coordination. Financial support was obtained from the Open Philanthropy Project (to MIT, Aarhus University, and SecureBio), an anonymous philanthropist from mainland China (to Tsinghua University), the Aphorism Foundation (to MIT and SecureBio), and Effective Giving (to MIT and SecureBio). The funders had no role in study design, data collection, data analysis, data interpretation, or writing of the report. To preserve international neutrality, no government funds were used to support this project.

Author Contributions

A.C.Y., D.G., K.M.E., L.F., and R.R. conceived the study; Y.Y., C.B., and M.G. wrote the cryptographic analysis with advice from A.C.Y., I.D., R.R., D.W., V.V., and A.S.; D.G. and K.M.E. created the figures, A.C.Y., M.G., Y.Y., C.B., I.D., K.L, J.L., X.L., Y.W., and H.Z. contributed to system design; software systems architecture & security was led by L.F; the software implementation was overseen by L.F. and J.B. and performed by L.F., J.B., T.V., M.K., W.C., F.S-L., L.V.H., S.W., and B.W-R.; database curation was led by T.V. and L.F. with input from K.M.E. and D.G.; performance assessments were performed by L.F. and D.G.; specificity analyses were conducted by D.G., L.F., T.V., and K.M.E.; and the certificate design and implementation were led by L.F. and F.S-L. All authors contributed to writing the paper.

Competing interests

K.M.E. and D.G. are authors of PCT/US2021/014814 filed by the Massachusetts Institute of Technology. All authors share an interest in preventing future pandemics.

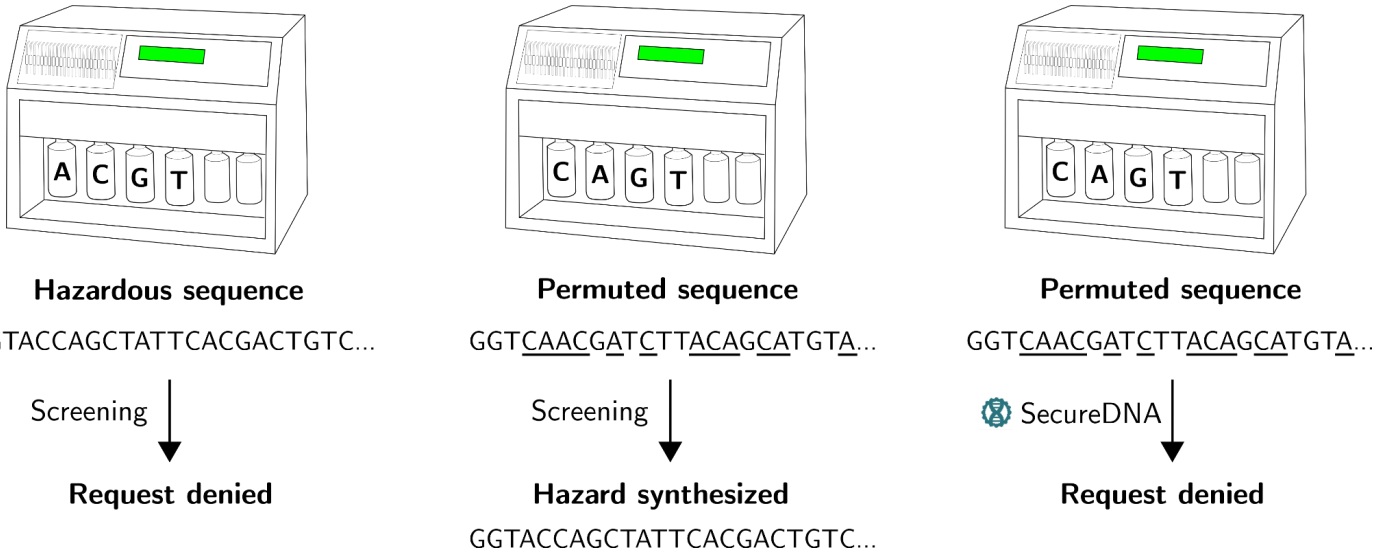
Data availability

The privacy of customer data from historical DNA synthesis orders is protected by legal agreements signed by the relevant providers. We encourage interested parties to reach out to providers directly to discuss the possibility of signing non-disclosure agreements.

Code availability

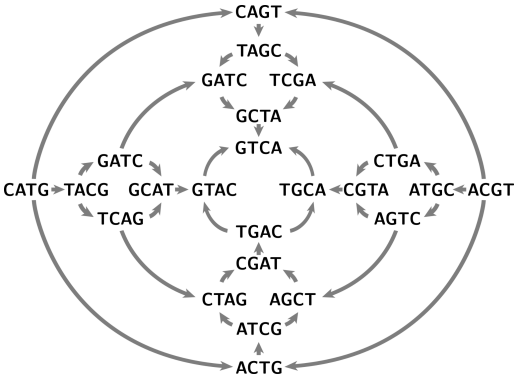
The open-source SecureDNA software is available at https://github.com/SecureDNA/.

a

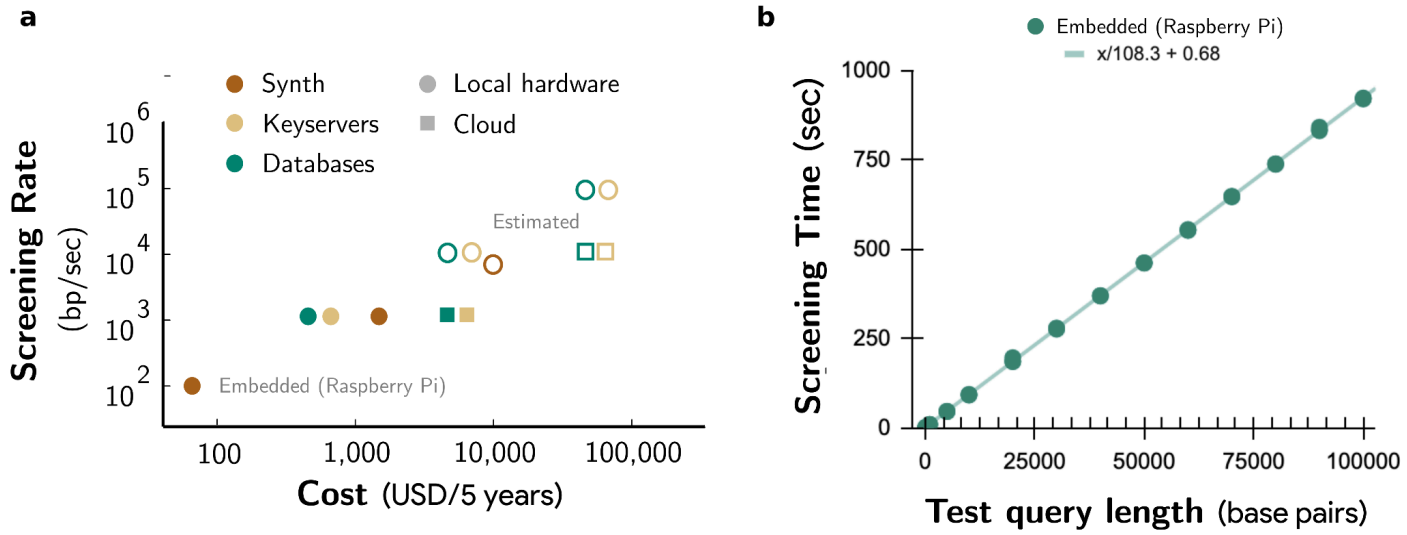


b

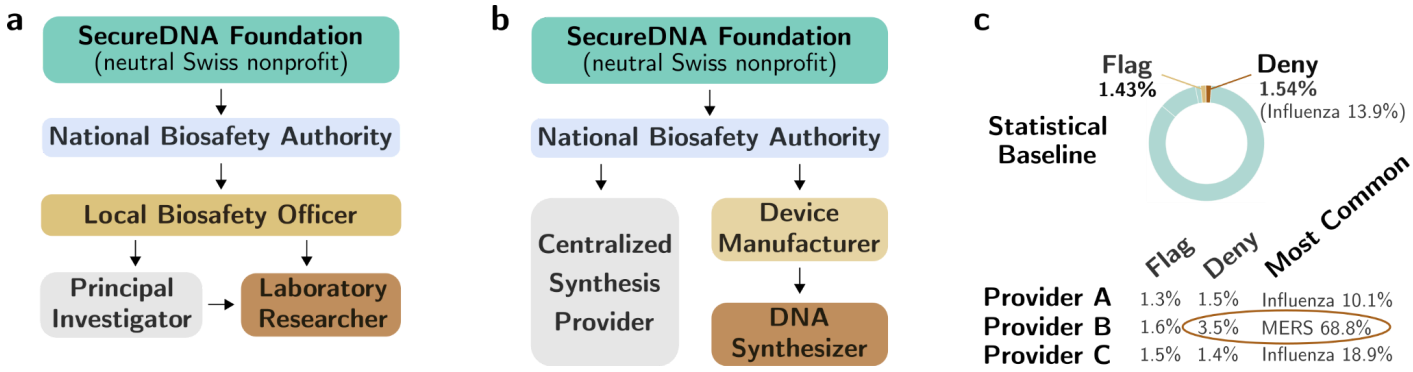
Permutation 1	Permutation 2	Permutation 3	Permutation 4
A ↔ C	-	-	-
A ↔ G	-	-	-
A ↔ T	-	-	-
C ↔ G	-	-	-
C ↔ T	-	-	-
G ↔ T	-	-	-
A ↔ C	G ↔ T	-	-
A ↔ G	C ↔ T	-	-
A ↔ T	C ↔ G	-	-
A ↔ C	C ↔ G	G ↔ A	-
A ↔ C	C ↔ T	T ↔ A	-
A ↔ G	G ↔ C	C ↔ A	-
A ↔ G	G ↔ T	T ↔ A	-
A ↔ T	T ↔ C	C ↔ A	-
A ↔ T	T ↔ G	G ↔ A	-
C ↔ G	G ↔ T	T ↔ C	-
C ↔ T	T ↔ A	A ↔ C	-
C ↔ T	T ↔ G	G ↔ C	-
A ↔ C	C ↔ G	G ↔ T	T ↔ A
A ↔ C	C ↔ T	T ↔ G	G ↔ A
A ↔ G	G ↔ C	C ↔ T	T ↔ A
A ↔ G	G ↔ T	T ↔ C	C ↔ A
A ↔ T	T ↔ C	C ↔ G	G ↔ A
A ↔ T	T ↔ G	G ↔ C	C ↔ A



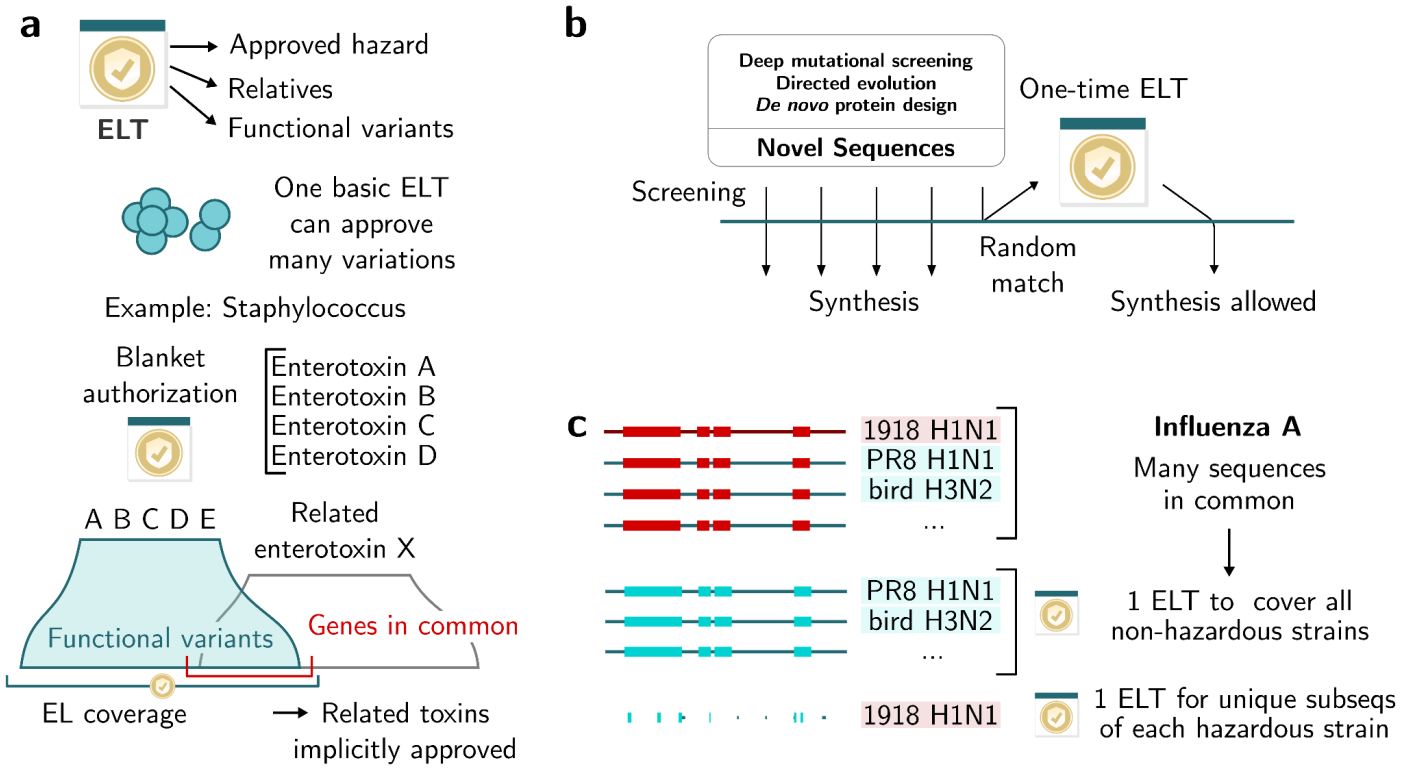
Extended Data Fig. 1 | Permutation attacks on benchtops. a) Many DNA synthesizers use four distinct reagents to add each of the four bases. Anyone with access to the machine can swap reagent bottles and permute the affected bases in their order to obtain the same DNA sequence. **b)** SecureDNA can screen benchtop queries for all 24 possible permutations of each subsequence to prevent reagent manipulation.



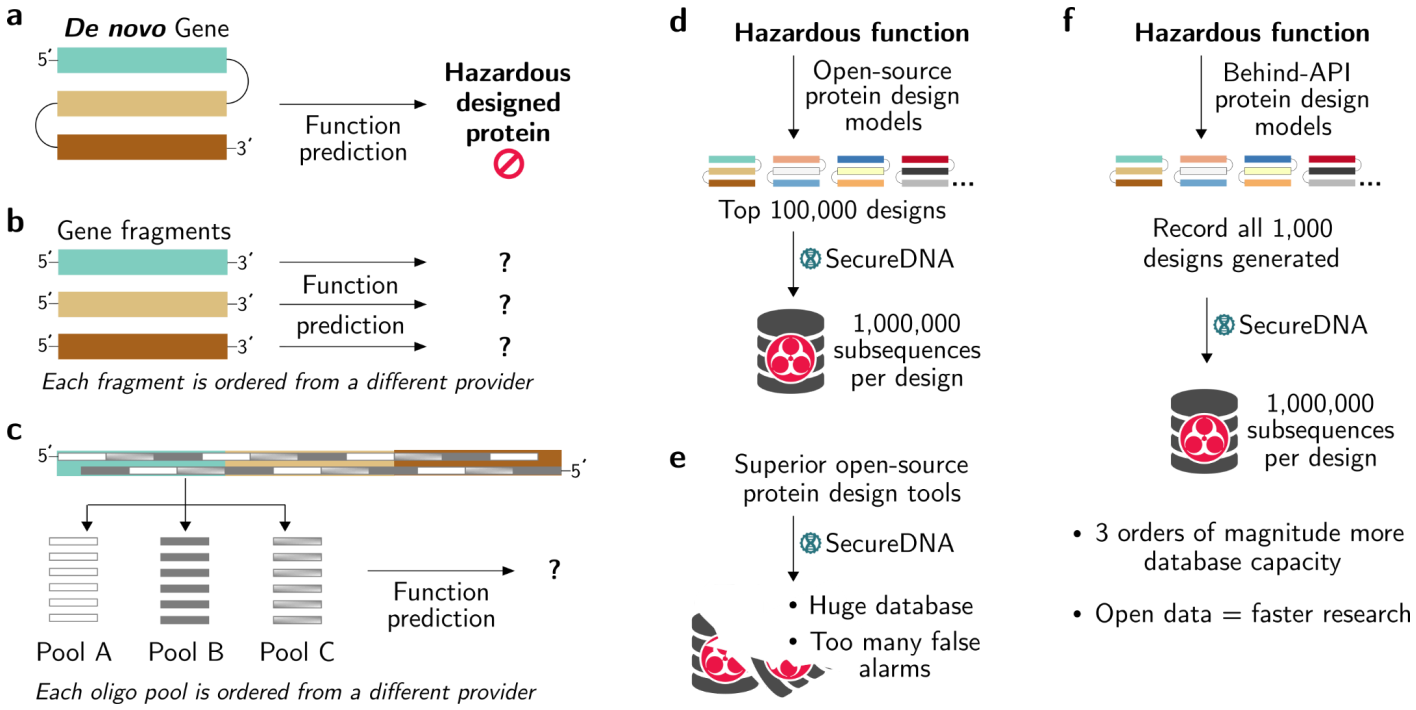
Extended Data Fig. 2 | Speed and cost of SecureDNA. a) Screening speeds of the synthesizer client, keyservers, and database as a function of hardware cost or cloud compute. Estimated values (open symbols) are extrapolated from prototype testing using diverse hardware and cloud-based implementations; hardware costs include redundancy for system robustness. **b)** Screening speed as measured by a Raspberry Pi implementation of synthclient running at greater than 100 bp/sec; more powerful processors yield correspondingly faster screening.



Extended Data Fig. 3 | SecureDNA certificate chains. a) The Switzerland-based SecureDNA Foundation issues exemption list certificates to each national biosafety authority, which in turn can issue certificates to local biosafety officers, which can issue certificates to principal investigators, and then to laboratory researchers. Researchers can either use a one-time exemption list token issued by their local biosafety officer or a laboratory exemption-list token together with the certificate issued to them by the lab's principal investigator. **b)** The SecureDNA Foundation issues screening certificates to each DNA synthesis provider and manufacturer of DNA synthesis machines. Each machine receives a certificate, which accompanies every screening order. **c)** The SecureDNA system records the number of matches to different public hazards associated with each certificate for analysis. Statistical associations concerning the number and pattern of matches to the hazard database can detect anomalous adversarial activity indicative of dictionary or other attacks.



Extended Data Fig. 4 | Exemption list versatility. a) A laboratory exemption list token (ELT) allows members to obtain any hazard approved by their local biosafety authority, as well as close relatives and predicted functional variants of that hazard. For example, a laboratory that works with staphyloccocal enterotoxins will receive an ELT linked to the primary accession numbers (ANs) of the relevant toxin-encoding genes of subtypes A through E. Because closely related toxins with different ANs would normally be recognized during screening by the subsequences they share with the primary ANs and predicted functional variants, the ELT gives access to all such toxins in the group. **b)** Oligonucleotide libraries for deep mutational scanning or directed evolution experiments and *de novo* designed genes do not correspond to sequences in repositories and may very rarely match a random hazardous subsequence. If this occurs, researchers can request a one-time exemption list token that will pass the specific set of sequences requested; because it is rare, harmless orders should not expect many such matches. **c)** For hazards such as influenza A, very few mutations separate harmless from hazardous strains. For example, most circulating strains share many subsequences with the 1918 H1N1 pandemic strain. Therefore, accessing any influenza sequences, even those from low-pathogenicity zoonotic influenza strains that cannot infect humans, requires a minimal "basic influenza A" ELT covering non-hazardous strains. However, this does not confer access to subsequences unique to the 1918 influenza virus and other hazardous strains, each of which requires a separate ELT.



Extended Data Fig. 5 | *De novo* designed proteins will eventually evade screening. a) Protein design tools will become increasingly capable of generating sequences with desired functions, including hazards. In principle, the same tool could predict the function of the generated sequence. **b)** In practice, these tools cannot be used for screening because hazardous designs can be generated in pieces with unpredictable folding and activity patterns. Dividing a designed sequence into three parts will generate three distinct polypeptide chains that are unlikely to fold into any structure with a predictable function. Ordering them from different providers at different times will preclude function prediction. **c)** Similarly, designed sequences can be assembled from pools of single-stranded oligonucleotides ordered from three separate providers, ensuring that no single provider can access the complete sequence, let alone predict its function. **d)** In principle, SecureDNA could defend most subsequences common to the top predicted hazardous sequences from leading public design tools. Since only a small fraction of DNA synthesis orders risk random false alarms because they are not present in repositories, it might be possible to generate a hazard database with as many as 10^{14} database entries while triggering only one random false alarm per month (Appendix D). Such a database could theoretically dedicate a million subsequences to defending each of 100,000 *de novo* designs for each of a thousand hazardous functions, although practical limitations may arise. **e)** Even given peptide screening, it will still be possible to generate designed hazards by assembling oligonucleotides too short for peptide screening, and the false alarm rate will gradually increase due to the need to defend more sequences as tools improve. Screening will eventually fail once enough *de novo* designs are possible for a given hazardous function. **f)** If protein design tools are only available through an API, and all designed sequences are logged, potentially hazardous designs can be included in the database and reliably detected.

	SecureDNA	Current alignment screening
Speed (asymptotic, Big O notation)	O(1); very fast	O(database size); slow
Privacy-preserving	Yes, cryptographically secure	No, too inefficient
Minimum window size	≥ 30 base pairs or 20 amino acids	≥ 200 base pairs (typical)
False alarm rate	Curation to remove harmless matches → no nonrandom false alarms	Many matches to unrelated genes require human review
Fully automatable	Yes	No, requires human review
Compatible with benchtop synthesizers/assemblers	Yes, given a secure connection	No, requires human review
Resistant to evasion by mutation or algorithmic design	Yes	No
Can screen for emerging hazards without disclosure	Yes[22]	No, requires disclosure; too inefficient to encrypt at scale[49]

Extended Data Table 1 | Characteristics of SecureDNA compared to current alignment-based screening approaches.

Appendix A: Maintaining an up-to-date database of hazards

If the hazards database is not kept up-to-date, adversaries keeping abreast of the literature and news will be able to immediately place orders and obtain newly credible pandemic viruses and other biological weapons. This is a major security vulnerability for all distributed solutions to DNA synthesis screening. SecureDNA solves this problem by maintaining a single database of subsequences from currently known hazards and updating it daily.

Specifically, SecureDNA staff maintain automated web alerts for potential pandemic pathogens and biological weapons. When a credible new agent is identified, all wild-type 30-mer subsequences are selected, encrypted via the keyservers, and the results added to the database without functional curation to provide immediate protection. Meanwhile, functional variants are predicted and curation is performed to generate subsequences capable of both sensitive and highly specific protection. Once encrypted, these new database entries replace the stopgap entries generated from the uncurated wild-type 30-mers. Once future red-teaming and prize competitions comprehensively assess the integrity of the SecureDNA architecture, it may be possible to establish a system to add emerging hazards that are not yet publicly credible to the database without disclosing their identities to anyone beyond the concerned researcher who flags the hazard and a single approved curator.

Similar alerts monitor the addition of novel threats to government lists of regulated hazards. When this occurs, the entries associated with an updated hazard are updated with suitable region tags to ensure compliance with local regulations and export control laws. If periodic literature reviews uncover evidence that an unregulated threat is no longer credible, the corresponding entries can be removed from the database.

Appendix B: A detailed description of the cryptography underlying SecureDNA

We now describe the cryptography behind the SecureDNA system using standard cryptographic terminology, although only on a high level. This description is aimed at a technical audience. Comprehensive technical details will be made available in forthcoming conference proceedings.

Overview

We consider the SecureDNA system in terms of different entities participating in the system. The three main entities are (1) the synthesizer; (2) the database server; and (3) the curator responsible for populating the database[1]. The synthesizer has inputs $S = \{s_1, \ldots, s_I\}$ which each are bit-strings of arbitrary length. The curator chooses a database $D = \{d_1, \ldots, d_J\}$, also consisting of bit-strings of arbitrary length. Finally, the database server obtains a special version of D, denoted as H. We require that the leakage of H about D must be kept to a minimum. The goal of SecureDNA is two-fold: we want the synthesizer and database server to engage in an interaction, at the end of which both learn $|S \cap D|$ without either learning any further information about D or S. Additionally, we require a protocol which allows the curator to generate H from D. To realize these goals, we assume additional entities participating in the protocol, namely n so-called keyservers. On a high level, the centerpiece of our solution is a so-called Distributed Oblivious Pseudorandom Function (DOPRF) [NPR99] where all n keyservers hold a share of a key k of a cryptographic hash function.

Notation

We write Z_p for the set $\{0, \ldots, p-1\}$ of residues of the integers Z modulo the prime p. The set of bit strings of arbitrary length is denoted as $\{0,1\}^*$. We assume that G is a finite abelian group of order p. G is considered in multiplicative notation and we write $\cdot : G \times G \to G$ to denote the group operation. For example, for any $g \in G$ we denote by g^2 the value $g \cdot g$ obtained from applying the group operation of G on g. We assume that the so-called Decision Diffie-Hellman [Boneh98] problem holds in the group G. In practice one can instantiate G e.g. using Groups over Elliptic Curves such as the well-known curve 25519 [Bernstein06].

[1] We consider the curator to be only an abstract entity and not a concrete organization. This will become more clear in the description below.

We assume the existence of a cryptographic hash function M which, on input from $\{0,1\}^*$, outputs a random element from the group G. Constructions for such hash functions towards groups such as G exist, such as e.g. Elligator [BHKL13].

Secret Sharing

We use a concept called Shamir's Secret Sharing [Shamir79]. It is parameterized by a number of parties n, a modulus p and a threshold $0 < t \leq n$. Further, Shamir's Secret Sharing uses an algorithm which, on input (n, p, t) as well as a secret $x \in Z_p$ creates shares $x_1, \ldots, x_n$ such that:

1. Anyone possessing fewer than t shares from $\{x_1, \ldots, x_n\}$ has no information about x.
2. Anyone possessing t or more shares of $\{x_1, \ldots, x_n\}$ can reconstruct x.

In Shamir's Sharing, the algorithm can be instantiated using polynomial arithmetic as follows:

1. To share the secret $x \in Z_p$, sample $t-1$ values $x_1, \ldots, x_{t-1}$ uniformly from Z_p.
2. Compute the unique monic degree-$t-1$ polynomial $f(X)$ with coefficients from Z_p where $f(0) = x$ and $f(i) = x_i$ for $i = 1, \ldots, t-1$.
3. Define $x_j = f(j)$ for $j = t, \ldots, n$

Lagrange Interpolation

Given any t shares (for simplicity, $x_1, \ldots, x_t$) there can only exist one monic polynomial f of degree $t-1$ with coefficients over Z_p such that $f(i) = x_i$ for $i = 1, \ldots, t$ by the fundamental theorem of Algebra. Using so-called Lagrange interpolation one can, using a set L of t evaluation points (in our example, $L = \{1, \ldots, t\}$) as well as an additional index h, compute coefficients $\lambda_1^{L,h}, \ldots, \lambda_t^{L,h}$ which are also from Z_p. Then, for these coefficients it must hold that $f(h) = \sum_{i \in L} \lambda_i^{L,h} x_i$ over Z_p. This means, one can evaluate the unique polynomial $f(X)$ in any point by computing a linear combination of any t evaluation points of f, and where the coefficients of the linear combination only depend on L, h but are independent of the concrete polynomial $f(X)$.

What is a DOPRF (in our setting)?

A DOPRF is an interactive protocol run between a client and n key holders. The client has an input $x \in \{0,1\}^*$ while each key holder has as input a Shamir share k_i of the key k. At the end of the interactive protocol, the client has learned a value $y \in G$ while no keyserver learns anything about x or y. Furthermore, the client learns nothing about k. Finally, for the value y it holds that it is uniformly random in G to anyone who does not know k, meaning that it reveals no information about x.

How is the DOPRF used in SecureDNA?

Upon system initialization, a random key k is chosen centrally and then secret-shared using the Shamir's scheme:, each of the n keyservers obtains a share of k, with keyserver i obtaining the share k_i. After this initial phase, the key k is securely deleted from the place where it was generated, such that only remaining information about it are the shares held by the keyservers.

After this initial phase is completed, the system is operational. The database server starts with an initially empty table H. To add values to it, the curator uses the DOPRF with inputs $d_1, \ldots, d_J$ together with the keyservers, obtaining the DOPRF outputs $hd_1, \ldots, hd_J$. It then sends $hd_1, \ldots, hd_J$ to the database server, which adds $hd_1, \ldots, hd_J$ to H This process can be repeated as often as necessary.

To test if a sequence is dangerous, the synthesizer will first compute $s_1, \ldots, s_I$ from the sequence and apply the DOPRF together with the keyservers to any s_i, computing hashes $hs_1, \ldots, hs_I$. It then sends the $hs_1, \ldots, hs_I$ to the database server, which informs the synthesizer if any hs_i shows up in H.

On the Security of SecureDNA

We assume that parties in SecureDNA only try to learn secrets from normally running the involved protocols, but they never actively deviate from the algorithms as specified[2]. This is the ``semi-honest" security model, which is well-established in cryptography. Given that the DOPRF has such semi-honest security itself, the same holds for our overall construction: the curator will not

[2] We can strengthen our security model to tolerate active corruption of the keyserver and database server. For the keyserver, one can protect each DOPRF computation using standard Zero-Knowledge proofs of exponentiation. To tolerate active corruption of the database server, one can replicate the database across multiple servers which are contacted by the synthesizer, who then uses the majority vote on their responses. For example, 3 copies can tolerate one corrupted server.

learn any information about k due to the security of the DOPRF, and the same holds for the synthesizer. At the same time, the keyservers learn no information about what the database is curated with or what a customer orders based on the security of the DOPRF. Finally, the database server never even talks to the keyservers, and the only values that it sees are uniformly random outputs of the DOPRF which reveal nothing about the inputs by assumption.

Additionally, we also have strong security of k: any $t-1$ or fewer corrupted (e.g. hacked) keyservers do not have enough information to recover k, based on the guarantees of the Shamir Sharing.

How the DOPRF is realized
SecureDNA uses a version of the NPR DOPRF (see [NPR99]):

1. On input $x \in \{0,1\}^*$ the client first chooses a uniformly random $\beta \in Z_p$ and computes $X = M(x)^\beta$. It then chooses a set $L \subset \{1,\ldots,n\}$ of t keyservers and sends (X, L) to each keyserver in L.
2. Each keyserver in L, upon obtaining (X, L) from the client, computes $\lambda_i^{L,0}$ as well as $Y_i = X^{k_i \cdot \lambda_i^{L,0}}$ and returns it to the client.
3. Upon obtaining all t responses from the keyservers in L the client computes and outputs $Y = (\prod_{i \in L} Y_i)^{\beta^{-1}}$

Here, the multiplication with β and β^{-1} in the exponent cancels out while the Lagrange coefficients interpolate the shares in the exponent, leading to the output being $Y = M(x)^k$. The value Y can be shown to be indistinguishable from a uniformly random element in the group G, due to the Decisional Diffie Hellman assumption assumed to hold in G. We refer to [NPR99] for more details about the security.

Observe that this version of the protocol is optimized for low computation on the client-side, as it outsources the application of the Lagrange coefficients to the servers while requiring the client to always obtain responses from all t keyservers in L. In case a keyserver i does not respond (e.g., downtime), the protocol can simply be restarted with a new set L that does not contain i.

Additional protection mechanisms for the key

In the final SecureDNA construction intended for release, we actually do not generate the key k centrally but use a distributed protocol such that key shares of k can be generated while k never appears on any machine. We additionally use this protocol to generate new keys $\tilde{k}$, and cryptographic protocols that allow us to generate an update key to update H from k to $\tilde{k}$ as well as a protocol for performing this update (without ever revealing the update key or H in the process). Moreover, we use so-called proactive secret sharing mechanisms which regularly redistribute k among the keyservers by generating new shares through a special cryptographic protocol. This means that, if someone steals key shares at time i, those cannot be used to reconstruct k any more after the key redistribution protocol was run, nor can any shares which span a redistribution event be used to reconstruct any keys. All of these protocols follow well-established design patterns for multiparty cryptography, such as outlined in [CDN15].

References for Appendix B

[Bernstein06] Bernstein, Daniel J. "Curve25519: new Diffie-Hellman speed records." *Public Key Cryptography-PKC 2006: 9th International Conference on Theory and Practice in Public-Key Cryptography, New York, NY, USA, April 24-26, 2006. Proceedings 9*. Springer Berlin Heidelberg, 2006.

[BHKL13] Bernstein, Daniel J., et al. "Elligator: elliptic-curve points indistinguishable from uniform random strings." *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. 2013.

[Boneh98] Boneh, Dan. "The Decision Diffie-Hellman problem." *International Algorithmic Number Theory Symposium*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998.

[CDN15] Cramer, Ronald, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[NPR99] Naor, Moni, Benny Pinkas, and Omer Reingold. "Distributed pseudo-random functions and KDCs." *International Conference on the Theory and Applications of Cryptographic Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.

[Shamir79] Shamir, Adi. "How to share a secret." *Communications of the ACM* 22.11 (1979): 612-613.

References

1. Neumann, G., Ozawa, M. & Kawaoka, Y. Reverse genetics of influenza viruses. *Methods Mol. Biol.* **865,** 193–206 (2012).
2. Maroun, J., Muñoz-Alía, M., Ammayappan, A., Schulze, A., Peng, K.-W. & Russell, S. Designing and building oncolytic viruses. *Future Virol.* **12,** 193–213 (2017).
3. Xie, X., Lokugamage, K. G., Zhang, X., Vu, M. N., Muruato, A. E., Menachery, V. D. & Shi, P.-Y. Engineering SARS-CoV-2 using a reverse genetic system. *Nat. Protoc.* **16,** 1761–1784 (2021).
4. Cello, J., Paul, A. V. & Wimmer, E. Chemical synthesis of poliovirus cDNA: generation of infectious virus in the absence of natural template. *Science* **297,** 1016–1018 (2002).
5. Esvelt, K. M. Delay, detect, defend: Preparing for a future in which thousands can release new pandemics. *Geneva Papers* (2022). at <https://dam.gcsp.ch/files/doc/gcsp-geneva-paper-29-22>
6. Mathieu, E., Ritchie, H., Rodés-Guirao, L., Appel, C., Giattino, C., Hasell, J., Macdonald, B., Dattani, S., Beltekian, D., Ortiz-Ospina, E. & Roser, M. Coronavirus Pandemic (COVID-19). *Our World in Data* (2020). at <https://ourworldindata.org/coronavirus>
7. Carroll, D., Daszak, P., Wolfe, N. D., Gao, G. F., Morel, C. M., Morzaria, S., Pablos-Méndez, A., Tomori, O. & Mazet, J. A. K. The Global Virome Project. *Science* **359,** 872–874 (2018).
8. Warren, C. J., Yu, S., Peters, D. K., Barbachano-Guerrero, A., Yang, Q., Burris, B. L., Worwa, G., Huang, I.-C., Wilkerson, G. K., Goldberg, T. L., Kuhn, J. H. & Sawyer, S. L. Primate hemorrhagic fever-causing arteriviruses are poised for spillover to humans. *Cell* **185,** 3980–3991.e18 (2022).
9. Sun, H., Li, H., Tong, Q., Han, Q., Liu, J., Yu, H., Song, H., Qi, J., Li, J., Yang, J., Lan, R., Deng, G., Chang, H., Qu, Y., Pu, J., Sun, Y., Lan, Y., Wang, D., Shi, Y., Liu, W. J., Chang, K.-C., Gao, G. F. & Liu, J. Airborne transmission of human-isolated avian H3N8 influenza virus between ferrets. *Cell* **186,** 4074–4084.e11 (2023).
10. Hou, Y. J., Chiba, S., Leist, S. R., Meganck, R. M., Martinez, D. R., Schäfer, A., Catanzaro, N. J., Sontake, V., West, A., Edwards, C. E., Yount, B., Lee, R. E., Gallant, S. C., Zost, S. J., Powers, J., Adams, L., Kong, E. F., Mattocks, M., Tata, A., Randell, S. H., Tata, P. R., Halfmann, P., Crowe, J. E., Jr, Kawaoka, Y. & Baric, R. S. Host range, transmissibility and antigenicity of a pangolin coronavirus. *Nat Microbiol* **8,** 1820–1833 (2023).
11. Thadani, N. N., Gurev, S., Notin, P., Youssef, N., Rollins, N. J., Ritter, D., Sander, C., Gal, Y. & Marks, D. S. Learning from prepandemic data to forecast viral escape. *Nature* **622,** 818–825 (2023).
12. Grange, Z. L., Goldstein, T., Johnson, C. K., Anthony, S., Gilardi, K., Daszak, P., Olival, K. J., O'Rourke, T., Murray, S., Olson, S. H., Togami, E., Vidal, G., Expert Panel, PREDICT Consortium, Mazet, J. A. K. & University of Edinburgh Epigroup members who wish to remain anonymous. Ranking the risk of animal-to-human spillover for newly discovered viruses. *Proc. Natl. Acad. Sci. U. S. A.* **118,** (2021).
13. Bügl, H., Danner, J. P., Molinari, R. J., Mulligan, J. T., Park, H.-O., Reichert, B., Roth, D. A., Wagner, R., Budowle, B., Scripp, R. M., Smith, J. A. L., Steele, S. J., Church, G. & Endy, D. DNA synthesis and biological security. *Nat. Biotechnol.* **25,** 627–629 (2007).
14. Diggans, J. & Leproust, E. Next Steps for Access to Safe, Secure DNA Synthesis. *Front Bioeng Biotechnol* **7,** 86 (2019).
15. Beal, J., Clore, A. & Manthey, J. Studying pathogens degrades BLAST-based pathogen identification. *Sci. Rep.* **13,** 5390 (2023).
16. Soice, E. H., Rocha, R., Cordova, K., Specter, M. & Esvelt, K. M. Can large language models democratize access to dual-use biotechnology? *arXiv [cs.CY]* (2023). at <http://arxiv.org/abs/2306.03809>
17. Nouri, A. & Chyba, C. F. DNA synthesis security. *Methods Mol. Biol.* **852,** 285–296 (2012).
18. Gretton D, Wang B, Foner L, Berlips J, Vogel T, Kysel M, Chen W, Sage-Ling F, Van Hauwe L, Wooster S, Weinstein-Raun B, DeBenedictis EA, Liu AB, Chory E, Cui H, Li X, Dong J, Fabrega A, Dennison C, Don O, Tong Y, Uberoy K, Rivest R, Gao M, Yu Y, Baum C, Damgard I, Yao AC, Esvelt KM. Random adversarial threshold search enables automated DNA synthesis screening. *SecureDNA project* (2024). at <https://www.securedna.org/download/Random_Adversarial_Threshold_Screening.pdf>
19. Shamir, A. How to share a secret. *Commun. ACM* **22,** 612–613 (1979).
20. Fowler, D. M. & Fields, S. Deep mutational scanning: a new style of protein science. *Nat. Methods* **11,** 801–807 (2014).
21. Packer, M. S. & Liu, D. R. Methods for the directed evolution of proteins. *Nat. Rev. Genet.* **16,** 379–394 (2015).

22. Biosafety Buyer's Guide - Biosafety Consultants. at <https://biosafetybuyersguide.org/consultants.html>
23. Lewis, T. The Quest to Overcome Gene Therapy's Failures. *Nature* (2021). doi:10.1038/d41586-021-02734-w
24. Carter, S. & DiEuliis, D. Mapping the Synthetic Biology Industry: Implications for Biosecurity. *Health Secur* **17,** 403–406 (2019).
25. DiEuliis, D. in *Cyberbiosecurity: A New Field to Deal with Emerging Threats* (ed. Greenbaum, D.) 71–78 (Springer International Publishing, 2023).
26. Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M. & Baker, D. De novo design of protein structure and function with RFdiffusion. *Nature* **620,** 1089–1100 (2023).
27. Alamdari, S., Thakkar, N., van den Berg, R., Lu, A. X., Fusi, N., Amini, A. P. & Yang, K. K. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv* 2023.09.11.556673 (2023). doi:10.1101/2023.09.11.556673
28. Chen, B., Cheng, X., Li, P., Geng, Y.-A., Gong, J., Li, S., Bei, Z., Tan, X., Wang, B., Zeng, X., Liu, C., Zeng, A., Dong, Y., Tang, J. & Song, L. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein. *arXiv [q-bio.QM]* (2024). at <http://arxiv.org/abs/2401.06199>
29. Baker, D. & Church, G. Protein design meets biosecurity. *Science* **383,** 349 (2024).
30. Office of the National Cyber Director. *Back to the Building Blocks: A Path Toward Secure and Measurable Software*. (The White House, 2024). at <https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf>