CellPress
OPEN ACCESS

Trends in
Biotechnology

Review

Cryptographic approaches to authenticating synthetic DNA sequences

Casey-Tyler Berezin[1], Samuel Peccoud[2], Diptendu M. Kar[3], and Jean Peccoud [1,4,5,6,*]

In a bioeconomy that relies on synthetic DNA sequences, the ability to ensure their authenticity is critical. DNA watermarks can encode identifying data in short sequences and can be combined with error correction and encryption protocols to ensure that sequences are robust to errors and securely communicated. New digital signature techniques allow for public verification that a sequence has not been modified and can contain sufficient information for synthetic DNA to be self-documenting. In translating these techniques from bacteria to more complex genetically modified organisms (GMOs), special considerations must be made to allow for public verification of these products. We argue that these approaches should be widely implemented to assert authorship, increase the traceability, and detect the unauthorized use of synthetic DNA.

Highlights

The ability to quickly and accurately verify the authenticity of synthetic DNA sequences is critical for cyberbiosecurity efforts.

Watermarks and digital signatures are the primary techniques for embedding attribution information into DNA sequences, and often implement error detection and correction to ensure robust communication.

Digital signatures provide integrity, authenticity, and non-repudiation assurances to facilitate the secure public verification of DNA sequences, and can even make DNA sequences self-documenting.

Features such as invisibility and zero-knowledge proofs may allow DNA signatures to be used to combat counterfeit genetically modified organisms (GMOs).

Machine learning approaches are being implemented to predict the source of unsigned DNA sequences.

Attribution of synthetic DNA mitigates cyberbiosecurity risks

In the 45 years since recombinant human insulin was first derived from plasmids expressed in *Escherichia coli*, the ease and availability of DNA synthesis has grown significantly [1,2]. Products utilizing synthetic DNA sequences (i.e., biological drugs, gene and cell therapies, biofuels, and foods like the 'Impossible Burger') have become the foundation of the growing bioeconomy [3–9]. Yet, since the inception of recombinant DNA technology, the cyber-physical nature of synthetic DNA has raised cyberbiosecurity concerns [5,9–13]. For example, stringent cyberbiosecurity protocols are needed to ensure DNA sequences encoding potentially dangerous products do not pass the security checks of DNA synthesis companies [14]. One major outstanding challenge is the ability to easily and accurately confirm the integrity and original attribution of DNA sequences [15].

After synthesis, DNA sequences are often manipulated and/or shared between academic laboratories. Thus, over time, owners and recipients may lack exact reference sequences and/or knowledge of the sequence origins. In addition, proper attribution of DNA sequences is limited by complicated patent regulations. The benefits of improved traceability and attribution of DNA sequences range from better quality control and the protection of intellectual property (IP) rights to the mitigation of security risks stemming from misuse and improved microbial forensics.

Since DNA sequences are generally designed and documented electronically, attribution may be given by including the origin information in the documentation. However, this only allows for an indirect association between the (digital) description of a product and its actual (physical) realization [5]. Furthermore, in some cases it can exacerbate security and IP risks: for example, if the full sequence of a pathogenic viral vector is freely available, or if a company wants to keep proprietary sequence information confidential [5,7].

Recent research has moved beyond digital documentation to within-molecule documentation. DNA is a robust storage medium capable of long-term data storage: books, songs, even operating

[1]Department of Chemical & Biological Engineering, Colorado State University, Fort Collins, CO, USA
[2]Department of Electrical Engineering, Colorado State University, Fort Collins, CO, USA
[3]Department of Computer Sciences, Northeastern University, Boston, MA, USA
[4]Department of Computer Sciences, Colorado State University, Fort Collins, CO, USA
[5]School of Biomedical Engineering, Colorado State University, Fort Collins, CO, USA
[6]Department of Systems Engineering, Colorado State University, Fort Collins, CO, USA

*Correspondence:
jean.peccoud@colostate.edu
(J. Peccoud).

© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

systems can be stored in DNA [16,17]. This review, intended for life science researchers interested in improving their knowledge of DNA cryptography, provides an overview of techniques for encoding encrypted information in DNA to ensure the traceability and attribution of synthetic DNA sequences. A discussion of early work on watermarking techniques, wherein short information-containing sequences are embedded into synthetic DNA sequences, provides a starting point for digital signature techniques, which provide both authenticity and integrity guarantees for an entire synthetic DNA sequence.

DNA watermarks can contain encrypted attribution information

One method for asserting ownership over digital media is watermarking, which involves embedding information in pictures, video, or audio. More recently, DNA watermarks have been developed, where a plaintext message (e.g., ownership information) is converted into a nucleotide sequence that can be extracted and verified (e.g., by PCR or sequencing). Jupiter *et al.* [18] recommended that all DNA watermarks be: (i) easily recoverable (stably integrated at a defined location), (ii) biologically innocuous (have no function of their own nor alter the function of the original DNA), (iii) error-tolerant (messages are recoverable in the event of unintentional errors), (iv) highly available (all laboratories have many unique sequences at their disposal), and (v) resistant to attack (complex and secure). Thus, DNA watermarks must simultaneously be long enough to be unique, complex, and information-rich, but also short enough that they are innocuous and economical.

To achieve these goals, principles from both coding theory and cryptography are applied. Coding theory aims to ensure that a message can be successfully encoded and transmitted over a noisy channel, and encompasses error detection and correction methods [19,20]. Cryptography, by contrast, aims to ensure secure, confidential communication only between specific parties (Box 1). Although watermarks do not inherently rely on encryption, here we focus primarily on instances where DNA watermarks are encrypted using a variety of cryptographic methods, from simple substitution ciphers to more complex public key cryptosystems, such as **Rivest–Shamir–Adleman (RSA)** (see Glossary) encryption.

In a landmark study, Clelland *et al.* developed a substitution cipher to encode alphanumeric messages in DNA (Figure 1A). Each character is encoded by a unique codon (e.g., 'M' is encoded by the nucleotides 'TTC'), and the entire message is flanked by PCR primer sequences at either end. After hiding the secret message in a pool of DNA fragments, the secret message can be successfully retrieved only by a recipient who has been given (i) the PCR primer sequences, and (ii) the sequence-to-character encryption key [21]. Wong *et al.* [22] expanded this system by transforming short (<100 bp) messages into bacteria to be replicated indefinitely. Importantly, the insert sequence is flanked by primer sequences which do not exist in the host genome, making it easily detectable. One must also note the landmark study by Gibson *et al.* [6] wherein they designed a synthetic *Mycoplasma mycoides* genome and differentiated it from the natural genome by embedding four unique watermark sequences. Although the watermarks were not designed to contain messages, the work provided a proof of principle for producing and tagging cells based on computer-designed genome sequences.

Once these early works showed that (hidden) information could be embedded in DNA and propagated in biological systems, methods that were more economical and/or optimized for use in living systems became desired. For example, Smith *et al.* adapted a **Huffman code** for use in DNA, wherein more frequently used characters are encoded by fewer nucleotides to shorten the message length (e.g., 'E' is encoded by T, while 'Z' is encoded by CCCTG); however, their method did not consider whether the message is biologically innocuous (Figure 1B) [23]. More recently, Zakeri *et al.* developed the iKey-64 system to map characters to codons while

Glossary

Convolutional neural network (CNN): a type of feed-forward neural network that leverages kernels or filters to process information locality. CNNs are well known for learning how to engineer features.

Designated confirmer signature: a cryptographic technique which is a type of digital signature that specifies a single confirmer or verifier when signed. The designated confirmer is then the only entity that can verify the signature, preventing others from doing so. This adds a level of privacy to digital signatures.

Discrete wavelet transform (DWT): a mathematical method which decomposes a signal or data series into its different frequencies. An advancement of the Fourier transform, this method provides information into frequency and temporal locality. Common applications are image compression, removing noise, and feature extraction.

ElGamal algorithm: a public-key encryption algorithm which leverages properties of modular exponentiation and the discrete logarithm problem to transmit data securely over insecure channels. It is used similarly to RSA.

Hamming code: a type of binary error detecting and correcting code. Hamming codes can be used to detect one- and two-bit errors and correct only one-bit errors. The detection and correction techniques work only for substitutions, not for insertions or deletions.

Hash function: a mathematical function extensively used in cryptography to calculate a fixed-size digest from input data. A key property is the computational infeasibility of reverse engineering or deducing the original input from its digest, thereby guaranteeing data integrity and enhancing security.

Huffman coding: a lossless data compression technique where a prefix Huffman code is inserted in the data to map common symbols to smaller symbols. This can be used in DNA by mapping frequently appearing characters in the data to fewer nucleotides.

Levenshtein distance: a type of edit distance metric that allows for flexible edit operations, including substitutions, insertions, and deletions. This flexibility makes it very applicable for genetic code.

Recurrent neural network (RNN): a type of artificial neural network that

Box 1. Fundamentals of cryptography

In information security, confidentiality, integrity, and availability make up the CIA triad that forms the basis of data protection [77]. Confidentiality ensures that data is accessed only by authorized entities (the goal of cryptography). Integrity allows recipients to verify that data has not been tampered with during transmission (error tolerance, a feature of coding theory).

Historically, encryption schemes were developed to securely transfer secret military messages. Ciphers are the earliest encryption schemes where a message is transformed, or encrypted, to conceal its meaning. In cryptography, the original human-readable message is called plaintext, and the concealed message is called ciphertext. Ciphers are reversible: encryption produces ciphertext from plaintext, while decryption reveals the original message from the ciphertext. For example, the popular Caesar cipher encrypts a message by shifting each letter in the original message by an index, concealing the message. If the index is two and the message is 'dna', 'd' is shifted by two and becomes 'f', 'n' becomes 'p', and 'a' becomes 'c'. The plaintext message 'dna' is thus transformed to the ciphertext 'fpc'.

Encryption schemes use one or more keys and a message as parameters. The Caesar cipher is a symmetric encryption scheme, wherein the same secret key (here, the index value of two) is used to decrypt the ciphertext in a simple reversal of the encryption process. Anyone who gains knowledge of the key can decrypt messages encrypted with the same key. Thus, it is difficult to share the key without compromising security.

To resolve this, asymmetric encryption, or public-key encryption, was developed using a pair of distinct keys that are mathematically linked: a public key and a private key. The public key is freely shared with anyone and is used for encrypting data, while the private key is kept secret and is used for decryption. When someone wants to send an encrypted message to the owner of the public key, they use that key to encrypt the data, and only the owner, who possesses the private key, can decrypt the original message, ensuring secure and confidential communication.

As messages became digitized for use in computers, the alphabet needed numerical representations: thus the American Standard Code for Information Interchange (ASCII) was introduced. ASCII maps every character and punctuation mark in the Latin alphabet to a number represented in binary code. The process of converting messages to ASCII and back is a type of encoding and decoding. With information now stored as numbers, encryption techniques shifted from human-friendly methods like the Caesar cipher to complex mathematical procedures that are computationally infeasible to break. There are a few mathematical concepts that are integral to cryptography, such as the modulo and XOR operation. The modulo operation (mod, %) calculates the remainder when one integer is divided by another (example: 5%3 = 2). The XOR (exclusive or) operation is a binary logical operation that returns true (1) if an odd number of its inputs are true, and false (0) otherwise.

preventing homopolymeric stretches of more than four nucleotides (e.g., GGGGG): more frequently used characters like 'E' are represented by three different nucleotides (e.g., TCA), while less common characters like 'K' may have two nucleotides repeated (e.g., TTC) [24]. In addition, the message is split across multiple DNA fragments, which can be engineered to contain decoy messages so that the true message can be interpreted only when a common primer is used to sequence a particular combination of fragments (Figure 1C).

Leier *et al.* [25] were the first to develop encrypted DNA watermarks using a binary (0 and 1) -system (Figure 1D). Short (26 bp) oligonucleotides were engineered to represent either start, stop, 0-bit, or 1-bit, and upon ligation of these DNA fragments, the resulting sequence encoded a string of numbers. After being mixed with dummy fragments – DNA from other species or fragments encoding other messages – the binary message can be decrypted by a user who knows (i) the secret key forward primer sequence, and (ii) the reverse primer sequences corresponding to the 0-bit and 1-bit sequences. The PCR products from both reactions are visualized on the gel to reconstruct the order of the bits. The message can be further encrypted by utilizing the dummy fragments themselves as the key: the gel image obtained from the dummy pool is subtracted from the gel image obtained from the encrypted pool (dummy pool + message strand) to reveal the message. Although this system is both secure and simple, the technical aspect of decryption is laborious, and the long length of each message makes this technique intractable on a large scale. Gehani *et al.* [26] described a system for encrypting binary messages in DNA watermarks utilizing a one-time pad (Figure 1E). Briefly, an 'exclusive or' (XOR or $\oplus$) calculation is performed using a single-use key which is the same length as the plaintext. An XOR calculation

maintains internal states as a form of memory, making the model effective for sequential datasets.

Reed–Solomon error-correction code: a type of binary erasure code that can detect and correct substitutions, insertions, and deletions.

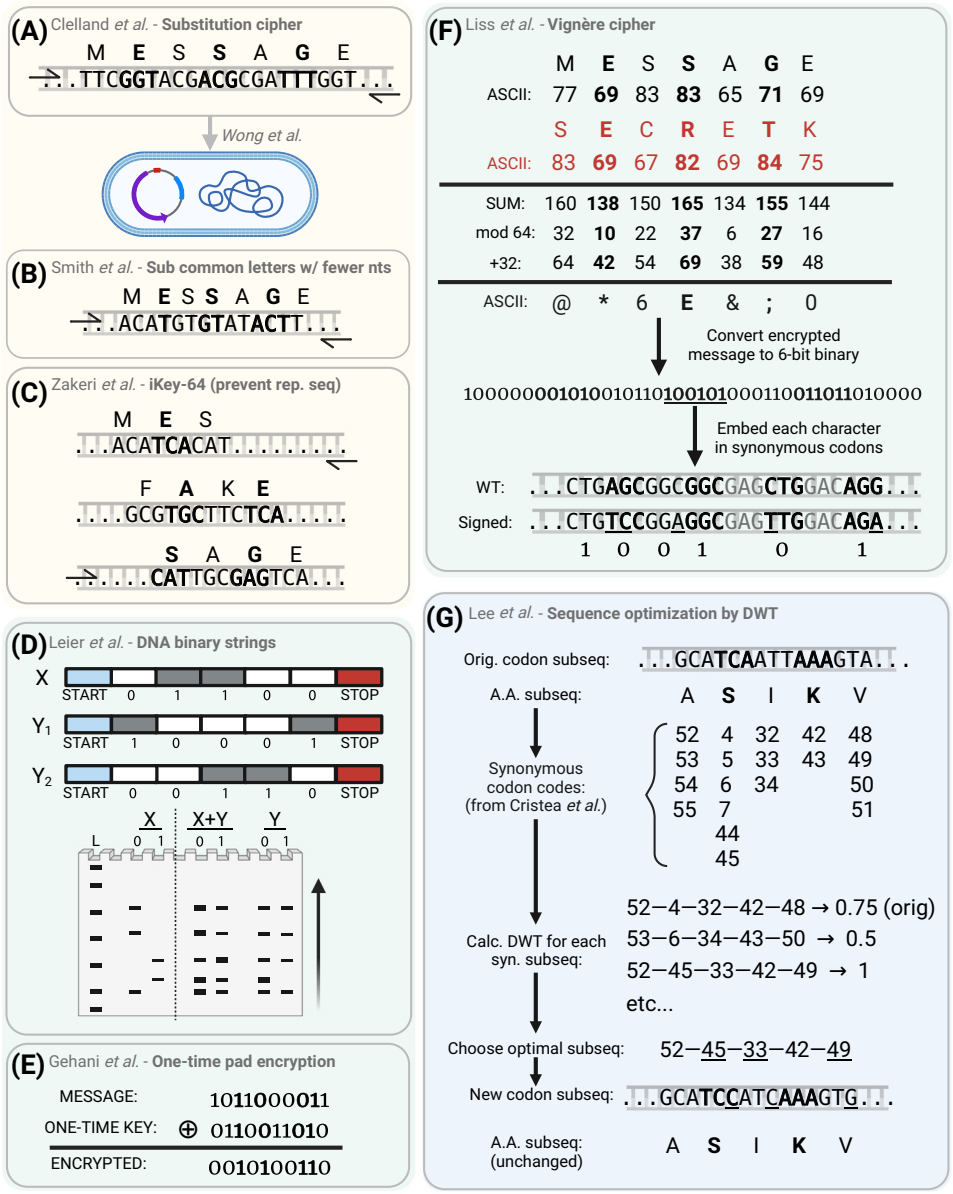
Rivest–Shamir–Adleman (RSA): a widely used cryptographic system that enables secure data transmission through asymmetric public-key encryption. The security of the system relies on the mathematical properties of large prime numbers. An alternative scheme to RSA would be the ElGamal algorithm.

Sakai–Kasahara signature scheme: this identity-based scheme encrypts a message for a given recipient with a known identity (like email or ORCID) so that it can only be decrypted by the recipient. This allows anyone to encrypt a message using only public information about the recipient.

Shamir's identity-based signature scheme: a digital signature that uses cryptographic techniques from RSA to secure a message and tie it to an identity. This scheme creates fixed-sized signatures that do not scale well for use in DNA signatures.

Vigenère cipher: this cipher encrypts an ASCII plaintext message using a changing Caesar cipher where each character has a different offset based on the key's character. The offset for the Caesar cipher is calculated from the key's value from the first value in the alphabet. The key's size is arbitrary and can be repeated to encrypt a message of any size.

Zero-knowledge (ZK) proof: a method that allows two parties, a verify and a prover, to assert the truth of some given information without revealing any of the information.



Trends in Biotechnology

Figure 1. Watermarking techniques allow information storage in DNA. (A–C) Substitution ciphers (yellow) encode messages in DNA by assigning each character to a short DNA sequence. Throughout, every other character of a message and its respective DNA/binary sequences are bolded for visibility. (A) DNA watermarks can be transformed into bacteria (Clelland *et al.* [21], [22]), (B) made more economical by decreasing the number of bases per character (Smith *et al.* [23]), and/or (C) made more secure by splitting the message across multiple fragments (Zakeri *et al.* [24]). (D–F) Methods using binary code (green) to encode messages in DNA add a layer of protection against interceptors. (D) DNA binary strings composed of START, STOP, 0-bit, and 1-bit sequences can be visualized on a gel. The message X is mixed with dummy fragments Y1 and Y2 (collectively Y) and the message is decrypted by subtracting the image of the dummy strands from the mixed image (X + Y). Each method relies on the interpreter knowing the key primer sequences (half arrows) and/or the 0- and 1-bit primer sequences (D), which produce fragments from START to each 0 or 1, which are read from the bottom up (Leier *et al.* [25]). (E) Binary messages can be encrypted with single-use keys. An 'exclusive OR' calculation is performed wherein different bits produce 0 and matching bits produce 1 (bolded) (Gehani *et al.* [26]).

(Figure legend continued at the bottom of the next page.)

is performed on each pair of bits in the plaintext message and key: a pair of two different bits (0 and 1) result in 0, while two of the same bit result in 1 (Figure 1E). If the key is used only once, this method is more secure than substitution-based methods. However, the key length scales with the message length; therefore, this method is not practical for encoding long messages.

Utilizing binary code as a conduit between a message and a DNA sequence presents a barrier to potential interceptors trying to read a DNA watermark. Embedding a watermark into a coding region of DNA may provide another level of concealment; however, it is more difficult to achieve than with DNA fragments or non-coding regions, since the underlying amino acid sequence cannot change. Liss *et al.* [27] encoded innocuous binary messages by only modifying amino acids that have synonymous codons: the first, third, and fifth common synonymous codons represent 1 (odds), whereas the second, fourth, and sixth most common ones represent 0 (evens). Therefore, if the interpreter knows the optimized codon usage table for a particular species, they can determine the binary message; thus, encrypting the message with a **Vigenère cipher** – a polyalphabetic substitution cipher – is critical. In brief, the American Standard Code for Information Interchange (ASCII) values of the plaintext message are added to the ASCII values of the secret key (Figure 1F). The encrypted ciphertext is generated through a few mathematical operations and made smaller by converting from typical 8-bit ASCII characters to a 6-bit binary code (Figure 1F). To decode the message, the process is followed in reverse, ending with the ASCII values of the key being subtracted from the watermark ciphertext. Since the receiver must know this secret key to interpret the watermark, decryption is more complex (secure) than with a simple substitution cipher.

Later, Haughton and Balado [28] developed BioCode, which facilitates watermarking in both non-coding and coding DNA regions using a binary-to-codon translation table. It ensures that no start codons are created by watermarks in non-coding regions, while in coding regions, it ensures that (i) the primary structure (translated protein) is conserved, and (ii) that the codon bias (usage) for a particular species is maintained. More recently, Lee [29] developed a **discrete wavelet transform (DWT)**-based method to optimize watermarks designed using the codon mapping table established by Cristea [30], wherein each codon is assigned to a number between 0 and 63 (Figure 1G). The codon sequence is separated into subsequences; then the DWT coefficients for each subsequence producing synonymous codons are calculated and used to choose the optimal subsequence for embedding the watermark. By choosing the optimal codon sequence, this approach was shown to be more robust to point mutations than BioCode [28] and the method by Liss [27], as well as DNA-Crypt [31] whose own merits are discussed later.

Error detection and correction in DNA watermarks

Innate to coding theory is the ability to successfully transmit a signal in the event of noise or errors. Given the inherent propensity of DNA to mutate, it is critical that DNA watermarking methods implement error detection, or better yet, correction techniques to ensure the attribution of DNA sequences in the long term. Special attention has been paid to error correction within short (>20 bp) barcodes for use in sequencing experiments (Box 2), but methods for longer DNA watermarks must take alternative approaches.

(F) Characters encoded by 8-bit American Standard Code for Information Interchange (ASCII) values can be encrypted with a secret key, converted into 6-bit binary sequences, and embedded only into synonymous codons (black) wherein 0 and 1 represent how common the codon is (Liss *et al.* [27]). (G) Discrete wavelet transform (DWT) coefficients can be calculated to find optimal synonymous codon subsequences (blue) (Lee *et al.* [29]). Figure created with BioRender.

Box 2. Error correction in sequencing barcodes

Short DNA barcodes (2–6 bp) for use in sequencing experiments were first developed in the late 1990s, such that different complementary DNA (cDNA) libraries could be tagged with and identified by unique, randomly-generated barcodes [78]. Qiu *et al.* then developed 6 bp error-correcting barcodes to tag individual cDNA molecules within multiplexed libraries [79]. This approach was based on the edit distance between two words (in this case, DNA sequences), which is the smallest number of changes needed to transform one string of characters into another (e.g., 'GCG' and 'GCA' have an edit distance of 1) [79,80]. Edit distance usually refers to Levenshtein distance, which is the most commonly used since it can correct not only substitutions, but insertions and deletions as well (e.g., 'GCG' and 'GC' have a Levenshtein distance of 1) [80,81]. Ashlock *et al.* [80] and Orth and Houghten [82] developed the Salmon algorithm which produces an optimized list of code words (barcodes) based on the Levenshtein distance and programmable biological constraints (e.g., GC content between 40% and 60%) [80,82]. Alternatively, Buschmann and Bystrykh [83] developed the sequence–Levenshtein distance for designing short barcodes: since the words in a DNA sequence (the barcode and the tagged sequence) are not separate, this method considers how the presence of the adjacent sequence will affect the edit distance of the barcode in the case of insertions and deletions.

However, since the edit distance is directly related to the length of the message, it is difficult to scale up these error correction methods to longer DNA messages; they are better suited for designing short (~6 bp) barcodes that correct only one or two errors. Even when edit distance is not used, a balance between error correction capability and barcode length must be met: Hawkins *et al.* produced a large list of 1 bp-correcting filled/truncated right end edit (FREE) barcodes that are between 3 and 16 bp long, and 2 bp-correcting barcodes that are between 5 and 17 bp long [84]. Nevertheless, these FREE barcodes can successfully correct insertions and deletions without knowing the length of the altered barcode (as would be common in a sequencing or ligation error), an advantage over the Levenshtein distance.

Early on, Smith *et al.* [23] implemented a comma code to detect errors in DNA watermarks: one nucleotide acts as a 'comma' that is consistently repeated throughout the watermark (Figure 2A). In this schema, the predictable pattern makes it easy to detect errors, and point mutations will introduce a stop codon 83% of the time. However, it greatly limits the information capacity of the message, since only three bases can be used for encoding. Arita and Ohashi [32] instead included a parity bit in the binary sequence of each character for error detection (Figure 2D). To insert their watermark, they mutated only the third nucleotide in each codon of a coding sequence, such that wobble base pairing can occur and the amino acid sequence is maintained. The wild-type codon encodes 0, whereas a mutated codon encodes 1. Each group of six bits (codons) encodes a character: the first five bits uniquely represent a character and the sixth bit is a parity bit for error detection (e.g., 'M' must be 011001 to confirm that the sequence has not changed). Thus, the integrity of the sequence can be verified upon sequencing if the interpreter knows the wild-type sequence.

Yamamoto *et al.* [33] developed a simple approach to retrieve watermarks even in the face of mutations. Message characters are first translated into short binary strings (e.g., 'M' is 100), then the binary strings are converted into nucleotides [i.e., 0 encodes a purine (A, G), while 1 encodes a pyrimidine (C, T)]. This system allows for more flexibility in the sequence that can be embedded within coding regions. Given the necessary primer sequences, the decoder can visually assess the integrity and degree of similarity between the obtained sequence and the watermark sequence on a DNA dot plot (Figure 2B). However, this method is labor-intensive, and can only be applied when the watermark sequence is known, not when embedding secret messages or in the case of wide public dissemination.

Several studies have implemented error correction techniques alongside cryptographic methods. The DNA-Crypt algorithm by Heider *et al.* [31,34] is based on a simple substitution cipher, where each base is represented by two binary digits (i.e., T: 00, G: 01, C: 10, A: 11), and the watermark is placed within synonymous codons as in [32]. Each byte includes four bits of information (two bases) plus four parity bits, so they can implement 8/4 **Hamming codes** to detect errors up to two bits and correct a single bit (Figure 2C). However, Hamming codes are only able to correct substitutions, not deletions or insertions (unlike the **Levenshtein distance**) (Box 2). Nevertheless,

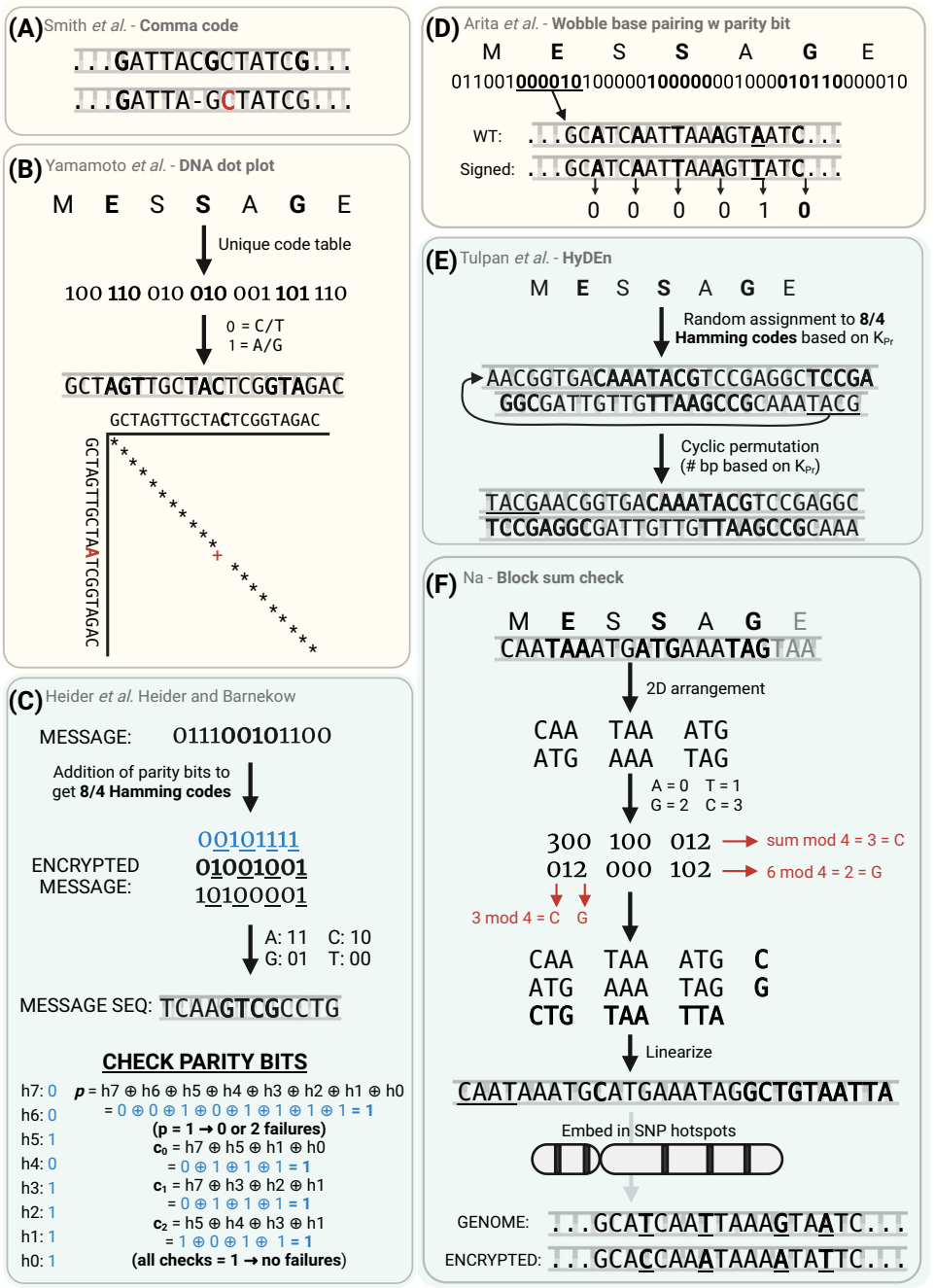


Figure 2. Error detection and correction techniques in DNA watermarks. (A) A predictable arrangement of one nucleotide makes it easy to confirm watermark integrity (Smith *et al.* [23]). (B) Discrepancies between a watermark sequence and an obtained sequence can be visualized on a DNA dot plot (Yamamoto *et al.* [33]). (D) The presence of a parity bit (bold) in the binary code of each character verifies sequence identity (Arita *et al.* [32]). While error detection can be rather simple (A,B,D; yellow), error correction techniques (C,E,F; green) are more involved. (C) Four information-containing bits are interspersed with four parity bits (underlined) in the construction of 1-base error-correcting 8/4 Hamming codes. The integrity of a resulting byte (blue) can be confirmed through a series of calculations on individual bits (h0–h7) (Heider and Barnekow [31]). (E) Hamming codes can be combined with additional encryption methods, such as

(Figure legend continued at the bottom of the next page.)

there are several algorithms to encrypt the message before embedding it. There are symmetric cryptosystems, in which the sender and receiver utilize the same (private) key for encryption and decryption, including one-time pad, advance encryption standard (AES), and Blowfish [35,36]. There are also asymmetric (public-key) cryptosystems, such as RSA, wherein the sender and receiver utilize different secret keys for encryption and decryption, increasing the security of the message [35,36].

A symmetric hybrid DNA encryption (HyDEn) approach by Tulpan *et al.* [37] also utilizes Hamming codes for error correction, but with an additional cyclic permutation to encrypt the message (Figure 2E). In brief, ASCII characters are randomly assigned to 8 bp binary DNA codewords, established based on the Hamming distance (e.g., 'M' is 'AACGGTGA'). After the message is converted to a DNA sequence, the message is encrypted by being permuted cyclically a certain number of positions based on a private key shared between the signer and recipient. This approach is less susceptible to a brute-force attack than simpler methods, such as a substitution approach.

A method by Na [38] encodes watermarks within single-nucleotide polymorphism (SNP) hotspots, defined as more than 35 SNPs in a 1 kilobase (kb) region (Figure 2F). This approach is fairly secure against unwanted third parties, because there is no obvious insert in the genome and SNPs are naturally polymorphic. Following use of a character-to-codon encryption table (e.g., 'M' is CAA), the codons are converted to a 4-bit binary string using a substitution system (A = 0, T = 1, G = 2, C = 3). A block sum check algorithm is then used for error correction. In brief, the resulting binary sequence is first arranged in 2D space. The values in each row and column are summed, divided by 4, and the remainder added to the end of the respective row or column (Figure 2F). The resultant parity values are converted back into nucleotides and inserted into the sequence. Upon decryption, the receiver can easily detect that the sequence has mutated if the obtained sequence does not produce the same parity nucleotides. Furthermore, if the obtained parity nucleotide is different, then one original nucleotide can be easily deduced and corrected based on the difference in the remainder. Of course, with the advent of the clustered regularly interspaced short palindromic repeats (CRISPR)–CRISPR-associated protein 9 (Cas9) system, the flexibility in where watermarks are inserted has grown [39].

From watermarks to digital signatures

Even when the messages within DNA watermarks are encrypted, there are inherent security risks in the use of watermarks [12,40]. Ultimately, a watermark is independent of the DNA vector in which it is embedded. Once DNA watermark sequences are identified, they can be easily deleted, edited, or engineered into new DNA molecules to obscure their contents or origins and potentially encode unwanted or harmful products [12]. Furthermore, the watermarking techniques that are based on substitution ciphers [21–24,26,28–30,32,33] are particularly prone to interception and forgery. The idiosyncrasies of the language of the encrypted message can make it easier to crack the message [37,41]. The Vigenère cipher used by Liss *et al.* [27] is slightly more secure, but the Kasiski method to guess the secret key length and decrypt the message was first published in 1863 [42,43]. In addition, symmetric encryption, in which both the sender and receiver

cyclic permutation, to increase the security of the message (Tulpan *et al.* [37]). (F) Groups of six characters can be arranged in 2D space, converted to base-four binary (0–3) and a block sum check for error correction performed. The sum of each column and row (after modulo 4) is converted to a nucleotide and appended to the linear DNA sequence, which is embedded into single-nucleotide polymorphisms (SNPs) in the genome. Decryption follows the encryption process in reverse, and a single error can be corrected based on the expected parity nucleotides for a certain row and column [38]. Figure created with BioRender.

share a secret key (e.g., primer sequence or encryption table) [21,22,25–27,31,33,37], is not practical in all cases. First, the watermark verification process is only possible within a preselected list of users who were given the key, and thus cannot be used in cases of further dissemination [40]. To share it with a new verifier, a new key must be generated and tracked. Second, an interceptor who gains access to the key can easily distort the encrypted message or generate new watermarked sequences. Furthermore, while the generation of many unique secret keys is trivial in the digital space, in the DNA domain each unique watermark must be synthesized and shipped, imposing significant financial and time constraints on the process.

Digital signatures, developed to mimic handwritten signatures, are widely used to validate the overall authenticity (i.e., identity, integrity, and origin) of digital documents, such as emails or financial transactions. They typically employ public-key (asymmetric) cryptographic methods, such as RSA, ensuring that signed items can be publicly and widely verified [12,44]. As opposed to watermarks, only one public key is needed to generate an encrypted signature that can be verified by anyone. The first notion of a digital signature was described by Diffie and Hellman [45], and over time, signatures have become equipped with stronger security and unforgeability guarantees. Consequently, researchers have adapted digital signatures for synthetic DNA. In essence, the sender uses a private key to generate a unique signature for the message, which is then attached to the message itself. The recipient then uses the sender's public key to verify the signature. If the signature is valid, it confirms that the message has not been tampered with (integrity) and was sent by the claimed sender (authenticity). Digital signatures thus enforce not only the CIA (confidentiality, integrity, and availability) triad, but also authenticate the data's origin and ensure non-repudiation, wherein the validity of the signature cannot be denied.

One approach, first described by Kar *et al.* [7,46] and experimentally validated by Gallegos *et al.* [47], uses identity-based signatures [48] to verify plasmid sequences (Figure 3). With identity-based signatures, users' secret signing keys are generated from their identifying information, such as an ORCID [49] or ID number, rather than being randomly generated. The public key is the identity (i.e., ORCID) itself. Anyone can derive their secret key from this public key using a trusted third party (TTP) server. As with other public key systems, if the authority is compromised, numerous user signatures can be easily forged. However, knowing the expected number of genuine keys can appease this issue.

The group initially used **Shamir's identity-based signature scheme**, which is based on RSA [50], to generate a 512 bp signature representing the entire document (plasmid). This means that the accuracy of not just the signature but the entire plasmid can be confirmed. The signature is produced from the message (a concatenation of the DNA sequence, plasmid ID, and signer's ORCID) using a SHA-256 **hash function**. Importantly, the signature was experimentally validated to not affect biological function (e.g., colony growth, antibiotic resistance, *lacZ* expression) [47]. The group then implemented the **Sakai–Kasahara signature scheme** [51] which generates a shorter 164 bp signature for any given input [47]. Both signature methods include a **Reed–Solomon error-correction code** (ECC) [52] in the signature cassette: it is 32 bp long, which facilitates the correction of two errors. However, the ECC can be made longer to correct more errors.

With this method, users can generate a unique digital signature cassette to verify a plasmid and its signer; however, the group also developed a method for inserting a fragment containing all the sequence annotations for the entire plasmid (Figure 3) [47]. Although increasing the number of annotations quickly makes the fragment quite large, a fragment containing minimal sequence

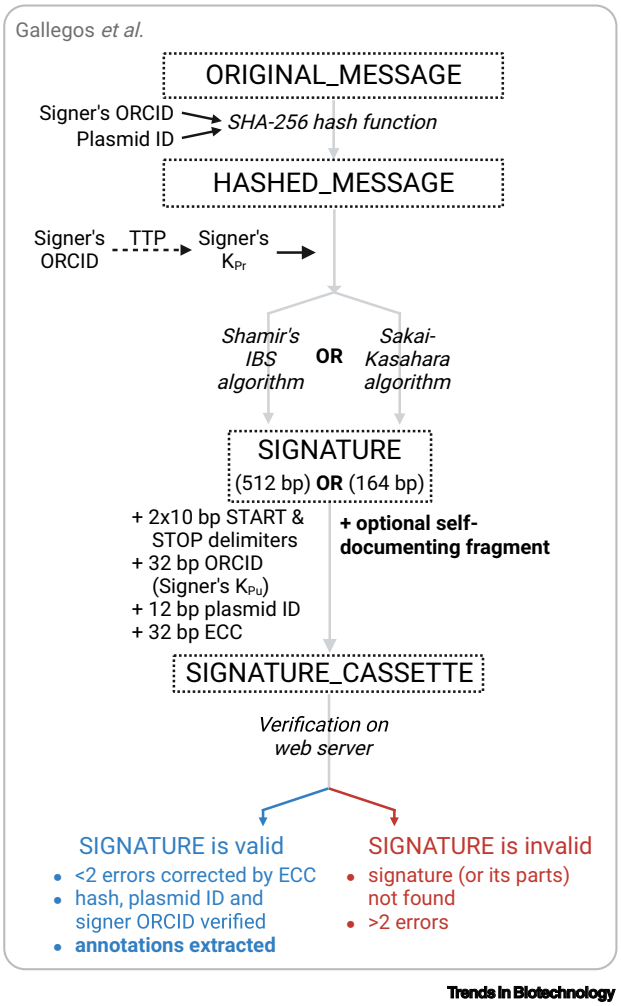


Figure 3. Digital signature techniques allow DNA sequences to be publicly verified. Signature techniques aim to securely encode more information into DNA than watermarks and are based around public-key cryptography methods that allow for wide, public verification of DNA sequences. An identity-based signature method developed for plasmids compresses the original message (the sequence itself) through one of two algorithms (resulting in two different-sized signatures) which utilizes the signer's private key, which is derived from their ORCID and issued by a trusted third party, to encrypt the hashed message. Additional information [e.g., plasmid ID, an error-correction code (ECC), or a self-documenting fragment encoding a Genbank file with gene annotations] is added to the signature cassette. The signature and plasmid itself are verified through a simple process on a publicly available web server (Gallegos *et al.*[47]). Figure created with BioRender.

annotations and a bibliographic reference to the source of the plasmid was successfully engineered into a plasmid using only a few thousand base pairs. This size could likely be decreased further with improved compression methods. Nevertheless, this approach allows a plasmid to be completely self-documenting: upon sequence assembly, the resulting FASTA file can be verified by the software, and the resulting Genbank file with the annotations can be obtained [47]. If the signed plasmid is later edited beyond the two errors corrected by the ECC, the signature is thus invalidated. Because the software is openly available for users to sign and verify plasmids, the process of obtaining all the necessary documentation to validate the origin, contents, and terms of use of the plasmid is quick and easy after sequencing assembly is complete.

A simpler approach to self-documentation has been implemented in a novel strain version control system, CellRepo, which uses a DNA barcoding system to generate digital twins between DNA sequences in cell strains and their digital documentation [53,54]. Rather than the barcode itself containing the identifying information, it points to the digital documentation hosted online where the information such as the sequence and origin can be confirmed. The inserted barcode comprises a sequencing primer binding site, a 96 bp randomly generated universally unique identifier

(containing the reference to the web documentation), a 9 bp synchronization sequence for aligning sequencing reads, and an 18 bp checksum sequence. Together, these components allow for error correction of the barcode, so it can be identified even with truncated or incorrect reads. Researchers can capture changes made to a cell strain in a new version, or 'commit', which prompts the insertion of a new barcode sequence into the strain. However, if the web documentation is not accessible for any reason, the barcode's usefulness is lost, as no other identifying information is directly embedded into the DNA (such as the signer's ID as in [47]).

Mueller [12] and Mueller *et al.* [40] describe the considerations for implementing digital signature techniques for securing entire GMOs. One important feature is a **zero-knowledge (ZK) proof**: (i) a confirmer can convince a verifier that they have the secret (a valid signature) without revealing it, and (ii) an interceptor cannot distinguish between a correct and an incorrect secret nor learn the secret [55,56]. This could be achieved through **designated confirmer signatures**, first introduced by Chaum in [57], which are verified by engaging in a usually interactive protocol with a TTP. Another important feature is invisibility: one cannot tell whether a signature is valid or not just by looking at it [12,58]. In the case of GMOs, this can be accomplished by designing two sets of signatures – valid and dummy signatures – that are embedded into different clones [12,40]. The hashes of all signatures can then be made publicly verifiable without revealing which are valid. This also allows illegitimate GMOs to be identified based on not having the full collection of valid signatures or the presence of modifications in dummy clones [12].

To address the specific challenges in signing GMOs, different cryptographic primitives (e.g., hashes, digital signatures) can be combined like building blocks [12,40]. For example, an asymmetric **ElGamal algorithm** [59,60] can be applied [40]. In brief, the sender's private key and the confirmer's public key are used to encrypt a message, which is decrypted using the confirmer's private key. If the verification process is successful, both the confirmer and the TTP are assured that the GMO is authentic, even though the exact signature is not known. In case of a dispute, the TTP can confirm the signature by sequencing. Therefore, it is not the verification of the signature itself that confirms the authenticity, but rather the outcome of the verification process (pass/fail). In addition, the actual signature is never disclosed, since it can be hashed and hidden in the genome by matching the codon bias of the organism [12,40]. However, the choice of primitives depends on the application. For example, the ElGamal algorithm produces an encrypted message twice the length of the message which will impose limitations on the signed message, and the interactive protocol required by designated confirmer signatures may not always be feasible. Error detection and correction techniques can also be integrated [40,61]. Although all new schemes need to be experimentally validated, the prospect of implementing ZK proofs and invisibility in signature schemes that can verify increasingly longer sequences is critical for securing GMOs. Despite the daunting logistics of implementing such signatures in GMOs at a large scale, these efforts are necessary to protect consumers and designers of products utilizing synthetic DNA sequences.

Concluding remarks

DNA watermarks and signatures allow information to be stably embedded in and retrieved from DNA sequences with varying levels of security and reliability (Figure 4, Key figure). In general, they rely on the relative simplicity of isolating short sequences or plasmids (up to thousands of base pairs long) from cells to confirm specific DNA sequences. In many cases, such verification is sufficient to fully authenticate the product of interest: it is typically more appropriate to verify the gene or plasmid of interest than an entire bacterial genome with base-level precision. Not only it is more time- and cost-efficient, but it is still technically difficult to sequence entire genomes with current sequencing technologies, due to repetitive or GC-rich regions that inhibit sequencing, systemic

Outstanding questions

How can DNA sequencing and assembly techniques be optimized to ensure robust retrieval of information encoded in DNA watermarks and signatures, given that the effectiveness of these methods ultimately depends on reliable data retrieval?

How can error correction methods for DNA signatures be improved, given the limitations of current techniques? In other words, how do we ensure error correction capabilities are appropriately scaled with the size of the DNA to correct the entire sequence, while also ensuring their size does not interfere with genetic functionality?

To what extent can cryptographic schemes for DNA signatures be standardized in the face of different biological requirements and applications (e.g., DNA data storage, bacterial plasmids used in research laboratories, viral vector vaccines, GMO crops and animals)?

How can the diverse documentation practices for genetic constructs, found in various repositories with unique structures and requirements, be standardized to facilitate the creation of formal data structures for programs generating embedded documentation? What specific information should these standards encompass: origin, purpose, gene annotations, experimental data?

Key figure

Watermark and digital signature methods vary in terms of security and reliability

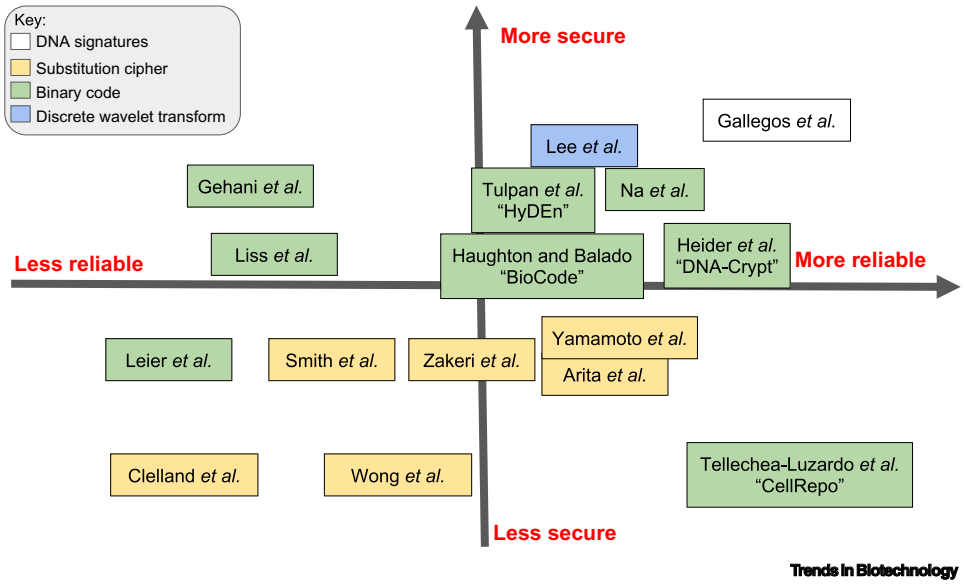


Figure 4. Security includes features such as the encryption and secrecy of the message, whereas reliability represents the ease with which the message can be retrieved (i.e., mutation resistance, error correction, availability of tools). Watermarks using substitution ciphers are generally insecure due to their simple nature (Clelland *et al.* [21], Wong *et al.* [22], Smith *et al.* [23], Zakeri *et al.* [24], Arita *et al.* [32], Yamamoto *et al.* [33]), while more complex systems like binary code (Leier *et al.* [25], Gehani *et al.* [26], Liss *et al.* [27], Haughton and Balado [28], Heider and Barnekow [31], Tulpan *et al.* [37], Na *et al.* [38], Tellechea-Luzardo *et al.* [53]) or discrete wavelet transform (Lee *et al.* [29]) increase the security of the message. Digital signatures are more secure than watermarks due to the use of more complex encryption algorithms and public key cryptography, but they vary in terms of their reliability (Gallegos *et al.* [47]). The presence of error correcting mechanisms is critical for long-term information storage in DNA.

errors introduced by PCR or sequencing, and errors introduced during the assembly of sequencing reads [62,63] (see Outstanding questions). In addition, *de novo* assembly efforts are inherently more difficult than reference-based assembly, in the event that a signature sequence is unknown before sequencing [64–66].

The idea that signatures are limited to verifying finite DNA sequences is especially pertinent as one moves into genetically modified plants and animals, as changes in unsigned regions of the genome could be missed [12,62]. However, it is unclear whether such long messages as genomic sequences can be signed in a way that is biologically innocuous and cost-effective. As sequences become longer, error detection and correction methods become even more crucial to the authentication of DNA messages. The implementation of HEDGES (hash encoded, decoded by greedy exhaustive search) error-correcting codes, which correct substitutions, insertions, and deletions, into current DNA signature schemes could prove fruitful [67].

Implementing signatures to verify GMOs is an important goal for the field, but further improvements are needed in both data encryption and sequencing methods to meet their specific challenges [12,40]. While data storage and compression in DNA encoding continue to improve [68], one

feasible approach may be to have multiple signatures each verifying a particular DNA sequence. The ability to verify even short sequences with signatures would be an improvement over current GMO verification methods that primarily rely on experimental quantification and characterization of the target sequence or protein [69,70]. Moving forward, cryptographic schemes must be adapted for a wide variety of biological applications – such as plasmids used for research, viral vector vaccines, and GMO crops – to protect creators and consumers alike. Ideally, these schemes could be standardized to an extent that would facilitate wide usage and streamlined signing and verification protocols.

Implementing DNA signatures in non-GMO higher living organisms could also be of benefit. Currently, DNA barcodes composed of random synthetic sequences are used to track and identify organisms. For example, recent work has engineered DNA-loaded protein microcrystals that can persist throughout a mosquito's lifespan [71]. Such DNA barcodes can be used to understand the circulation of vector-borne diseases or track endangered species, but these efforts would be strengthened by applying the cryptographic methods discussed here to embed meaningful identifying information within these barcodes.

In addition to watermarks and signatures, which aim to explicitly label DNA with identifying information, machine learning techniques are now emerging to predict the source laboratory or designer of non-labeled synthetic DNA sequences (Box 3). The premise is that the unique design choices made (consciously or not) by research groups (e.g., preferences for certain genetic parts, codon optimization, and the propagation of silent mutations) collectively leave a 'methodological signature' which may be used to infer attribution of the DNA construct [15,72]. Since these approaches do not rely on sequences being embedded in the DNA sequence, there are no concerns about modifying or losing the properties of the DNA as with watermarking or signature methods. However, for the same reason, these approaches do not provide authenticity

Box 3. Machine learning for genetic engineering attribution

Nielsen *et al.* [72] trained a **convolutional neural network (CNN)** on a dataset of over 30 000 plasmid sequences from over 800 laboratories. The depositing laboratory was predicted correctly 48% of the time and placed in the top-10 predictions 70% of the time. Importantly, accurate predictions could still be made even if, on average, 316 point mutations were made (about 10% of the sequence). Nevertheless, the effect of mutations greatly depends on the plasmid or genetic part: in one case, a 12 bp disruption in a region containing two restriction sites completely changed the predicted source laboratory.

Whereas CNNs are primarily used for pattern recognition in spatial data (i.e., usually images, sometimes text), **recurrent neural networks (RNNs)** are better suited for detecting and predicting patterns in sequential data (i.e., text, as in natural language processing models) [72,73]. Thus, the deteRNNt model established by Alley *et al.* achieved an accuracy of about 70% and the source laboratory was in the top-10 predictions nearly 85% of the time, using a larger dataset of over 80 000 plasmid sequences from over 3500 laboratories [73].

A recent community-led approach led to even higher accuracy predictions [85]. Using a similar dataset as Alley *et al.* [73], 75 teams surpassed the top-10 accuracy of both of the existing models [72,73]. The top-ranked team achieved nearly 95% top-10 accuracy using a CNN-based approach, and the misclassification rate for the top-1 prediction was less than 20% (compared with 30% in [73] and 50% in [72]). In addition, one of the teams did not use neural networks, but rather a naive Bayes classifier, resulting in a nearly 1000-fold increase in speed.

Machine learning models require high-quality, representative training datasets, are computationally expensive to run, and may have low explainability (how and why the model makes a decision). In response to these limitations, Wang *et al.* [86] developed an alternative alignment-based approach. Using dataset similar to that of Nielsen *et al.* [72], they constructed a pan-genome composed of all the identified genomic regions. To identify a plasmid's origin, its subsequences are aligned against the pan-genome, and a laboratory score, favoring more unique sequences, is calculated. This approach led to an accuracy of 76% and the source laboratory was in the top-10 predictions 85% of the time. Although this method is still limited by the contents of the training dataset, it is easier to incorporate new sequences into the pan-genome than to re-train a machine learning model.

guarantees. There is little to stop third parties from mimicking the methodological signature of a research group to conceal a construct's origin [15]. Conversely, the presence of a feature that is commonly associated with one laboratory does not mean that that laboratory created that construct. Although these models will likely never achieve 100% accuracy, they will be useful in providing a starting ground for investigations into the origins of DNA sequences [72,73].

DNA signatures are a promising way of embedding identifying information in synthetic DNA sequences – such as their source, designer, terms of use, or even a compressed copy (hash) of the sequence itself – and then allowing public verification of the entire sequence. In line with efforts to make synthetic DNA design more transparent and standardized, the contents of DNA signatures must be standardized [73–76]. For example, the use of the ORCID as a standardized identification number is appealing due to its prevalence among scientific researchers. Though neural networks have shown some potential for identifying the originating laboratory of DNA sequences, they sorely lack the precision necessary for reliable authentication and ultimately fulfill a different need than DNA signatures. Rather than trying to computationally predict who designed a plasmid, researchers in the field must move towards signing synthetic DNA sequences themselves. As the use of DNA signatures becomes more widespread, the lack of a signature could be an indication in and of itself that the sequence is dubious.

Acknowledgments

J.P. and C-T.B. are supported by the National Science Foundation (award #2123367) and the National Institutes of Health (R01GM147816).

Declaration of interests

J.P. and S.P. have financial interests in GenoFAB, Inc. This company may benefit or be perceived as benefiting from this publication.

References

1. Hughes, R.A. and Ellington, A.D. (2017) Synthetic DNA synthesis and assembly: putting the synthetic in synthetic biology. *Cold Spring Harb. Perspect. Biol.* 9, a023812
2. Goeddel, D.V. *et al.* (1979) Expression in *Escherichia coli* of chemically synthesized genes for human insulin. *Proc. Natl. Acad. Sci. U. S. A.* 76, 106–110
3. Voigt, C.A. (2020) Synthetic biology 2020–2030: six commercially available products that are changing our world. *Nat. Commun.* 11, 6379
4. Chiarabelli, C. *et al.* (2013) Chemical synthetic biology: a mini-review. *Front. Microbiol.* 4, 285
5. Peccoud, J. *et al.* (2018) Cyberbiosecurity: from naive trust to risk awareness. *Trends Biotechnol.* 36, 4–7
6. Gibson, D.G. *et al.* (2010) Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* 329, 52–56
7. Kar, D.M. *et al.* (2018) *Digital signatures to ensure the authenticity and integrity of synthetic DNA molecules*, Proceedings of the New Security Paradigms Workshop, August 2018, pp. 110–122
8 Shankar, S. and Hoyt, M.A. (2018) Impossible Foods Inc. Expression constructs and methods of genetically engineering methylotrophic yeast. US Patent #US9938327B2
9. Walsh, P.F. (2022) "Securing the bioeconomy: exploring the role of cyberbiosecurity". In *The Handbook of Security* (Gill, M., ed.), pp. 335–355, Springer
10. Murch, R.S. *et al.* (2018) Cyberbiosecurity: an emerging new discipline to help safeguard the bioeconomy. *Front. Bioeng. Biotechnol.* 6, 39
11. Mueller, S. (2021) Facing the 2020 pandemic: what does cyberbiosecurity want us to know to safeguard the future? *Biosaf. Health* 3, 11–21
12. Mueller, S. (2019) On DNA signatures, their dual-use potential for GMO counterfeiting, and a cyber-based security solution. *Front. Bioeng. Biotechnol.* 7, 189
13. Berg, P. *et al.* (1974) Potential biohazards of recombinant DNA molecules. *Science* 185, 303
14. Puzis, R. *et al.* (2020) Increased cyber-biosecurity for DNA synthesis. *Nat. Biotechnol.* 38, 1379–1381
15. Lewis, G. *et al.* (2020) The biosecurity benefits of genetic engineering attribution. *Nat. Commun.* 11, 6294
16. Church, G.M. *et al.* (2012) Next-generation digital information storage in DNA. *Science* 337, 1628
17. Erlich, Y. and Zielinski, D. (2017) DNA Fountain enables a robust and efficient storage architecture. *Science* 355, 950–954
18. Jupiter, D.C. *et al.* (2010) DNA watermarking of infectious agents: progress and prospects. *PLoS Pathog.* 6, e1000950
19. Aydin What is coding theory and what is cryptography?, *Introduction to Coding Theory and Cryptography*. Kenyon College, OH, USA https://www2.kenyon.edu/Depts/Math/Aydin/Teach/Sp09/328/Intro.pdf
20. Calderbank, A.R. (1998) The art of signaling: fifty years of coding theory. *IEEE Trans. Inf. Theory* 44, 2561–2595
21. Clelland, C.T. *et al.* (1999) Hiding messages in DNA microdots. *Nature* 399, 533–534
22. Wong, P.C. *et al.* (2003) Organic data memory using the DNA approach. *Commun. ACM* 46, 95–98
23. Smith, G.C. *et al.* (2003) Some possible codes for encrypting data in DNA. *Biotechnol. Lett.* 25, 1125–1130
24. Zakeri, B. *et al.* (2016) Multiplexed sequence encoding: a framework for DNA communication. *PLoS ONE* 11, e0152774
25. Leier, A. *et al.* (2000) Cryptography with DNA binary strands. *Biosystems* 57, 13–22
26. Gehani, A. *et al.* (2004) DNA-based cryptography. In *Aspects of Molecular Computing: Essays Dedicated to Tom Head, on the*

Occasion of His 70th Birthday (Jonoska, N. *et al.*, eds), pp. 167–188, Springer
27. Liss, M. *et al.* (2012) Embedding permanent watermarks in synthetic genes. *PLoS One* 7, e42465
28. Haughton, D. and Balado, F. (2013) BioCode: two biologically compatible algorithms for embedding data in non-coding and coding regions of DNA. *BMC Bioinformatics* 14, 121
29. Lee, S.-H. (2014) DWT based coding DNA watermarking for DNA copyright protection. *Inf. Sci.* 273, 263–286
30. Cristea, P.D. (2002) Conversion of nucleotides sequences into genomic signals. *J. Cell. Mol. Med.* 6, 279–303
31. Heider, D. and Barnekow, A. (2007) DNA-based watermarks using the DNA-Crypt algorithm. *BMC Bioinformatics* 8, 176
32. Arita, M. and Ohashi, Y. (2004) Secret signatures inside genomic DNA. *Biotechnol. Prog.* 20, 1605–1607
33. Yamamoto, N. *et al.* (2014) A watermarking system for labeling genomic DNA. *Plant Biotechnol.* 31, 241–248
34. Heider, D. and Barnekow, A. (2008) DNA watermarks: a proof of concept. *BMC Mol. Biol.* 9, 40
35. Patil, P. *et al.* (2016) A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish. *Procedia Comput. Sci.* 78, 617–624
36. Simmons, G.J. (1979) Symmetric and asymmetric encryption. *ACM Comput. Surv. (CSUR)* 11, 305–330
37. Tulpan, D. *et al.* (2013) HyDEn: a hybrid steganocryptographic approach for data encryption using randomized error-correcting DNA codes. *Biomed. Res. Int.* 2013, 1–11
38. Na, D. (2020) DNA steganography: hiding undetectable secret messages within the single nucleotide polymorphisms of a genome and detecting mutation-induced errors. *Microb. Cell Factories* 19, 128
39. Velázquez, E. *et al.* (2021) Targetron-assisted delivery of exogenous DNA sequences into *Pseudomonas putida* through CRISPR-aided counterselection. *ACS Synth. Biol.* 10, 2552–2565
40. Mueller, S. *et al.* (2016) A covert authentication and security solution for GMOs. *BMC Bioinformatics* 17, 1–8
41. Shiu, H.-J. *et al.* (2010) Data hiding methods based upon DNA sequences. *Inf. Sci.* 180, 2196–2208
42. Bhateja, A.K. *et al.* (2015) Cryptanalysis of vigenere cipher using cuckoo search. *Appl. Soft Comput.* 26, 315–324
43. Kasiski, F.W. (1863) *Die Geheimschriften und die Dechiffrir-Kunst. Mit besonderer Berücksichtigung der deutschen und der französischen Sprache*, Sprache E.S. Mittler
44. Salomaa, A. (2013) *Public-Key Cryptography*, Springer
45. Diffie, W. and Hellman, M. (1976) New directions in cryptography. *IEEE Trans. Inf. Theory* 22, 644–654
46. Kar, D.M. *et al.* (2020) Synthesizing DNA molecules with identity-based digital signatures to prevent malicious tampering and enabling source attribution. *J. Comput. Secur.* 28, 1–31
47. Gallegos, J.E. *et al.* (2020) Securing the exchange of synthetic genetic constructs using digital signatures. *ACS Synth. Biol.* 9, 2656–2664
48. Baek, J. *et al.* (2004) *A survey of identity-based cryptography*. Proc. of Australian Unix Users Group Annual Conference
49. Haak, L.L. *et al.* (2012) ORCID: a system to uniquely identify researchers. *Learned Publ.* 25, 259–264
50. Shamir, A. (1984) *Identity-based cryptosystems and signature schemes*, Workshop on the theory and application of cryptographic techniquesSpringer
51. Sakai, R. and Kasahara, M. (2003) ID based cryptosystems with pairing on elliptic curve. *IACR Cryptol. ePrint Arch.* 54, 1–6
52. Reed, I.S. and Solomon, G. (1960) Polynomial codes over certain finite fields. *J. Soc. Ind. Appl. Math.* 8, 300–304
53. Tellechea-Luzardo, J. *et al.* (2020) Linking engineered cells to their digital twins: a version control system for strain engineering. *ACS Synth. Biol.* 9, 536–545
54. Tellechea-Luzardo, J. *et al.* (2022) Versioning biological cells for trustworthy cell engineering. *Nat. Commun.* 13, 765
55. Goldreich, O. and Oren, Y. (1994) Definitions and properties of zero-knowledge proof systems. *J. Cryptol.* 7, 1–32
56. Goldwasser, S. *et al.* (1989) The knowledge complexity of interactive proof systems. *SIAM J. Comput.* 18, 186–208
57. Chaum, D. (1994) Designated confirmer signatures. In *Workshop on the Theory and Application of Cryptographic Techniques*, Springer
58. Galbraith, S.D. and Mao, W. (2003) *Invisibility and anonymity of undeniable and confirmer signatures*, Topics in Cryptology – CT-RSA 2003: The Cryptographers' Track at the RSA Conference 2003 San Francisco, CA, USA, April 13–17, 2003 ProceedingsSpringer
59. ElGamal, T. (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* 31, 469–472
60. Rivest, R.L. *et al.* (1978) A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* 21, 120–126
61. Mueller, S. *et al.* (2015) Improving dependability and precision of data encoding in DNA. *Eur. J. Exp. Biol.* 10,
62. Mueller, S. (2019) Are market GM plants an unrecognized platform for bioterrorism and biocrime? *Front. Bioeng. Biotechnol.* 7, 1–14
63. Hu, T. *et al.* (2021) Next-generation sequencing technologies: an overview. *Hum. Immunol.* 82, 801–811
64. Liao, X. *et al.* (2019) Current challenges and solutions of *de novo* assembly. *Quant. Biol.* 7, 90–109
65. Prjibelski, A. *et al.* (2020) Using SPAdes de novo assembler. *Curr. Protoc. Bioinformatics* 70, e102
66. Wick, R.R. *et al.* (2017) Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Comput. Biol.* 13, e1005595
67. Press, W.H. *et al.* (2020) HEDGES error-correcting code for DNA storage corrects indels and allows sequence constraints. *Proc. Natl. Acad. Sci. U. S. A.* 117, 18489–18496
68. Wang, C. *et al.* (2022) Mainstream encoding–decoding methods of DNA data storage. *CCF Trans. High Perform. Comput.* 4, 23–33
69. Fu, W. *et al.* (2020) A universal analytical approach for screening and monitoring of authorized and unauthorized GMOs. *LWT* 125, 109176
70. Qian, C. *et al.* (2018) Recent advances in emerging DNA-based methods for genetically modified organisms (GMOs) rapid detection. *TrAC Trends Anal. Chem.* 109, 19–31
71. Stuart, J.D. *et al.* (2022) Mosquito tagging using DNA-barcoded nanoporous protein microcrystals. *PNAS Nexus* 1, pgac190
72. Nielsen, A.A. and Voigt, C.A. (2018) Deep learning to predict the lab-of-origin of engineered DNA. *Nat. Commun.* 9, 1–10
73. Alley, E.C. *et al.* (2020) A machine learning toolkit for genetic engineering attribution to facilitate biosecurity. *Nat. Commun.* 11, 6293
74. Peccoud, J. *et al.* (2011) Essential information for synthetic DNA sequences. *Nat. Biotechnol.* 29, 22
75. Martínez-García, E. *et al.* (2022) SEVA 4.0: an update of the Standard European Vector Architecture database for advanced analysis and programming of bacterial phenotypes. *Nucleic Acids Res.* 51, D1558–D1567
76. Czar, M.J. *et al.* (2009) Writing DNA with GenoCAD™. *Nucleic Acids Res.* 37, W40–W47
77. Samonas, S. and Coss, D. (2014) The CIA strikes back: redefining confidentiality, integrity and availability in security. *J. Inf. Syst. Secur.* 10, 21–45
78. Bonaldo, M.d.F. *et al.* (1996) Normalization and subtraction: two approaches to facilitate gene discovery. *Genome Res.* 6, 791–806
79. Qiu, F. *et al.* (2003) DNA sequence-based 'bar codes' for tracking the origins of expressed sequence tags from a maize cDNA library constructed using multiple mRNA sources. *Plant Physiol.* 133, 475–481
80. Ashlock, D. *et al.* (2012) On the synthesis of DNA error correcting codes. *Biosystems* 110, 1–8
81. Levenshtein, V.I. (1966) *Binary codes capable of correcting deletions, insertions, and reversals*, Soviet physics doklady, Soviet Union
82. Orth, J. and Houghten, S. (2011) Optimizing the Salmon Algorithm for the construction of DNA error-correcting codes. *IEEE* 1–7
83. Buschmann, T. and Bystrykh, L.V. (2013) Levenshtein error-correcting barcodes for multiplexed DNA sequencing. *BMC Bioinformatics* 14, 272
84. Hawkins, J.A. *et al.* (2018) Indel-correcting DNA barcodes for high-throughput sequencing. *Proc. Natl. Acad. Sci. U. S. A.* 115, E6217–E6226
85. Crook, O.M. *et al.* (2022) Analysis of the first genetic engineering attribution challenge. *Nat. Commun.* 13, 7374
86. Wang, Q. *et al.* (2021) PlasmidHawk improves lab of origin prediction of engineered plasmids using sequence alignment. *Nat. Commun.* 12, 1167