AMERICAN BINARY

Version: 1.0.1 | **Date:** 27 Dec 2024

Ambit Client Cryptography FAQ

American Binary, Inc.

11335 NE 122nd Way Suite 105

Kirkland, WA 98034

e: support@ambit.inc | w: ambit.inc

AMERICAN BINARY

1. Contents

AMERICAN BINARY

1. The Advent of Quantum Computing

1.1. What is quantum computing?

Classical computers store and manage information as a function of binary digits, or bits. Bits exist in one of two states, 0 or 1, which are represented electrically within circuits as either an on or an off state. Quantum computers store and manage information as a function of quantum bits, or qubits. A qubit is like a bit in that it can have a state of 0 or 1. At the same time, qubits are not at all like bits in that they are composed of atoms or subatomic particles, and they can represent **all states between** 0 and 1 taken at once. This property is called **superposition**.

As a result, a qubit can store dramatically more information than a classical bit and a quantum computer can process massive amounts of information concurrently and many orders of magnitude faster than a classical computer while using significantly less energy.

1.2. Quantum vs Classical Computation Space Requirements

Perfect Qubits	Complex Numbers in State Vector	Classical Bits Needed	Classical Memory Size
1	2	32	4 bytes
2	4	64	8 bytes
3	8	128	16 bytes
4	16	256	32 bytes
5	32	512	64 bytes
10	1,024	16,384	2 KB
20	1,048,576	16,777,216	2 MB
30	1,073,741,824	17,179,869,184	2 GB
40	~1 trillion	~17 trillion	2 TB
50	~1 quadrillion	~17 quadrillion	2 PB
100	~1.27 × 10^30	~2 × 10^31	2.5 × 10^18 TB
1,000	~1.07 × 10^301	~1.72 × 10^302	~2.15 × 10^289 TB
1,000,000	~1.07 × 10^301,030	~1.72 × 10^301,031	~2.15 × 10^301,018 TB
10,000,000	~1.07 × 10^3,010,300	~1.72 × 10^3,010,301	~2.15 × 10^3,010,288 TB

Notes:

- Assumes 16 bytes per complex number (double precision)
- Classical bits needed = Number of complex numbers × 16 bytes × 8 bits/byte
- These numbers assume perfect qubits with no error correction overhead

- For perspective, estimated atoms in observable universe ≈ 10^80
- This demonstrates the math of a 'perfect' aka "fully error corrected qubits in a gate-model quantum computer" (not an annealer)

1.3. What is cryptography?

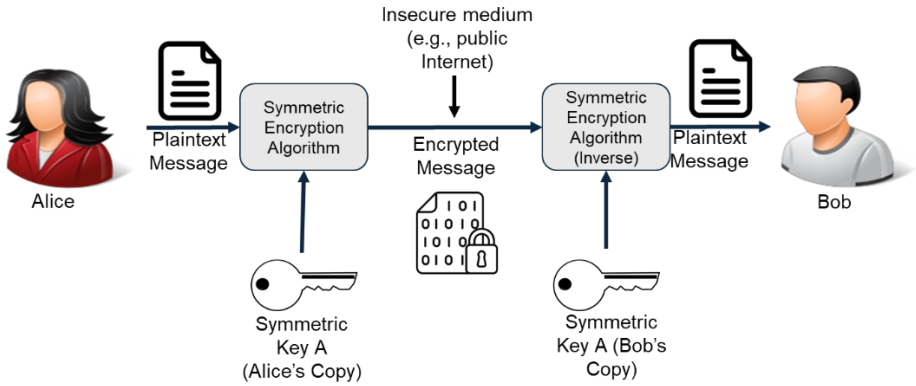
Cryptography is the study and practice of altering information such that only the intended recipient or recipients can read it. Cryptography enables parties to a communication to send information across insecure channels in a manner that prevents unauthorized parties from reading the communication's contents. Modern digital cryptography provides four principal guarantees:

- Confidentiality: Only authorized users can access the encrypted information
- Integrity: Parties to the communication can verify that the information contained in the communication has not been altered by an unauthorized party
- Authenticity: Parties to the communication can verify that the information exchanged is from the stated source
- Non-repudiation: Parties to the communication cannot deny that they sent or received a message or took other digital actions

1.4. What are symmetric and asymmetric cryptography?

There are four components to a cryptographic message exchange:

- The **plaintext** is the unencrypted form of the communication, readily readable to anyone who possesses it.
- The **cryptographic algorithm** is the mechanism by which the plaintext is encrypted.
- The **key(s)** is(are) the instruction set that tells the cryptographic algorithm how to encrypt and/or decrypt the plaintext.
- The **ciphertext** is the encrypted version of the plaintext that's produced when the plaintext is run through the encryption algorithm pursuant to the instructions embodied in the key.



There are two types of cryptographic algorithms by which plaintext can be encrypted to produce ciphertext, and by which ciphertext can be decrypted to produce the original plaintext message:

- **Symmetric, or secret, key algorithms**
- **Asymmetric, or public, key algorithms**

The difference between the two comes down to how many keys are maintained by each party to the communication.

In a symmetric cryptographic algorithm, the **same key** is used to encrypt and decrypt the message, and both the sender and recipient must possess a copy of the key. In this case, the key, which is the only sensitive information in the process, is established and exchanged in advance of the transmission of any information across the insecure channel (e.g., the public internet) so that both parties possess a copy. The sender composes the message and then processes it through the symmetric cryptographic algorithm according to the instructions contained in the key. The resulting ciphertext is then sent across the insecure medium to the recipient. The recipient processes the received ciphertext through the symmetric cryptographic algorithm using the **same** key to produce the original plaintext.

In an asymmetric cryptographic algorithm, each party to the exchange has **two** cryptographic keys: A **public key** that is used to encrypt the plaintext for a specific recipient, and a mathematically related **private key** that is used to decrypt the resulting ciphertext. This set of mathematically related keys is called a **key pair**. Importantly, the instruction set contained in the public key cannot be used to derive the private key and/or decrypt the ciphertext. (The use of a public key is why asymmetric cryptography is often referred to as **public key cryptography**.) As a result, it can be freely exchanged across insecure channels (and even posted publicly) without compromising the security of the cryptographic transmission.

When transmitting information using asymmetric cryptography, each party derives a key pair independently and transmits their public key to the other party while retaining their private key and keeping it confidential. To reiterate, neither party ever transmits their private key. The sender then processes the plaintext through the asymmetric cryptographic algorithm with the instructions contained in the recipient's public key and transmits the resulting ciphertext to the recipient across an insecure channel (e.g., the public internet). Upon receiving the ciphertext the recipient processes it through the asymmetric cryptographic algorithm using their private key.

The use of a public-private keypair enables asymmetric cryptography to solve the **key distribution problem**; that is the transmission of key material across insecure channels like the public internet.

1.5. How widespread is cryptography in the technological landscape?

Cryptography is, in a word, everywhere. It is the critical enabler for nearly every technology that enables modern life, from web browsing to online financial transactions to instant messaging, credit cards, debit cards, wireless key fobs for automobiles to, multifactor authentication tokens. Instruction sets for cryptographic algorithms are manufactured into the chips that power servers as well as desktop and

mobile devices. Cryptographic management technology is found in almost every browser in use today. If it's not the most pervasive technology, it's certainly among the top tier. The modern connected world would simply be unable to function without it.

1.6. Why is asymmetric cryptography important and what does it enable??

Asymmetric cryptography, as described in section 2.3, is a cryptographic system in which each party has two keys, a public key, which is used to encrypt information for transmission to a specific recipient, and a private key, which is used by the recipient to decrypt the information encrypted with the public key. Because the public key is not sensitive and gives away nothing with respect to its corresponding private key, the public key can be freely shared over insecure channels such as the public internet. As a result, asymmetric cryptography solves what is known as the **key distribution problem**.

The two-key structure of an asymmetric cryptographic system enables it to be used to provide several security guarantees. As noted in section 2.2, these include confidentiality, integrity, authenticity, and non-repudiation. Security mechanisms that depend on asymmetric cryptography include authentication mechanisms used by computer systems, automobile security systems, credit cards, and debit cards, digital certificates, digital signatures, and key exchange.

1.7. What impact does quantum computing have on asymmetric cryptography?

The security provided by asymmetric cryptography relies on a class of mathematics known as **trapdoor functions**. Examples of trapdoor function mathematics include **integer factorization** and **the discrete logarithm problem**. A trapdoor function is easy to compute in one direction, but difficult to compute in the opposite direction without special information. In this case, *difficult* relates to the amount of time that would be required for a classical computer (e.g., a workstation, a notebook, a smartphone) to back-calculate the inputs from which the trapdoor function had created an output. Typically, this level of computational difficulty is measured in tens, if not hundreds of thousands of years.

As noted in section 2.1, quantum computing enables massively concurrent processing yielding exponential acceleration in computation. As a result, it is anticipated that through the use of currently known computing mechanisms (e.g., **Shor's Algorithm**, **Grover's Algorithm**, etc.), quantum computers will be able to solve the trapdoor functions in minutes to days as opposed to tens or hundreds of thousands of years, thus rendering contemporary asymmetric cryptography obsolete.

1.8. When will we start to feel the impact of quantum computing on asymmetric cryptography?

Despite the emergence of stable, viable quantum computing platforms not being expected until close to the end of the decade (near to 2030), quantum computing is already having an indirect impact on asymmetric cryptography. Governments and malicious actors around the world are currently engaging in what is known as **Harvest-Now-Decrypt-Later (HNDL)** attacks. HNDL attacks are premised on a combination of the following ideas:

AMERICAN BINARY

- Information transiting the internet is routinely protected with asymmetric cryptographic algorithms that cannot be broken with classical computers today;
- Quantum computing will be able to break these asymmetric algorithms;
- While much of the information that transits the internet loses value quickly over time, a large, significant part of the information retains value over long periods; and
- Seemingly innocuous information transiting the internet from many sources can be mosaicked to create valuable sensitive information.

As a result, global actors are currently copying and storing everything that transits the internet and are doing so **today**.

1.9. How does Ambit Client help organizations prepare for the advent of quantum computing?

Ambit Client defeats Harvest-Now-Decrypt-Later (HNDL) attacks through the use of post-quantum cryptography (**PQC**) that is compliant with the Federal Information Processing Standard (FIPS) 140-3 and Commercial National Security Algorithm Suite (CNSA) 2.0 standards. When using Ambit Client, all information leaving an endpoint (e.g., workstation, laptop, mobile device) is fully encrypted with PQC. While bad actors may be able to capture information, they will not be able to decrypt, use, or derive value from it. As a result, Ambit Client creates future-proofing for a post-quantum world.