

Entrechat: A Strictly Synchronous Peer-to-Peer Messaging System

Melik Lemariey

M.Sc. Telecommunications Systems

Independent Researcher

melik.e.lemariey@proton.me

2025

Abstract

This document presents Entrechat v1, a strictly synchronous peer-to-peer messaging system, designed around deliberately restrictive architectural constraints.

Entrechat excludes servers, accounts, directories, asynchronous delivery mechanisms, message storage, and any persistent infrastructure. These exclusions are not accidental limitations, but explicitly assumed design choices.

The system does not claim any universal security properties. The guarantees described are contextual, dependent on clearly stated assumptions, and limited to a defined scope of use. Entrechat favors architectural clarity and control of operational exposure over functional exhaustiveness.

Keywords: peer-to-peer messaging, strict synchrony, security architecture, synchronous communication, Tor, end-to-end encryption, operational exposure

1 System Scope

Entrechat is a mobile messaging system designed for intentional and explicit exchanges between previously identified peers.

The system is defined by the following structural constraints:

- strictly peer-to-peer communication,
- strictly synchronous exchanges,
- absence of servers or operated backend,
- absence of accounts, directories, or discovery mechanisms,
- absence of asynchronous delivery and message storage,
- absence of multi-device replication.

Network transport relies exclusively on the use of the Tor network [4]. Identities are cryptographic, generated locally and managed locally by users.

This document describes Entrechat v1 as implemented. No future evolution or functional extension is implied.

2 Purpose and Usage Context

Entrechat addresses a deliberately narrow need: to enable direct and strictly synchronous communication between mutually available peers, without relying on an operated service or persistent infrastructure.

The system was initially designed based on a developer’s own need, considered as a particular case revealing constraints encountered in contexts of punctual exchanges with strong exposure-control requirements.

Entrechat is not designed as a general-purpose messenger. It neither aims to replace existing platforms nor to cover the full spectrum of contemporary uses. It explores a specific design space in which:

- communication occurs only when both parties are present,
- sessions have explicitly bounded lifecycles,
- no implicit network activity is generated,
- architectural complexity is deliberately limited.

Synchrony is considered a positive property. It makes availability constraints explicit and avoids the introduction of implicit mechanisms such as queues, deferred delivery, or off-session metadata persistence.

Entrechat neither seeks invisibility nor protection in case of endpoint compromise. These aspects are explicitly considered out of scope.

3 Epistemological Position and Usage Assumptions

The security properties of a software system are inherently contextual and dependent on the assumptions retained. A threat model is a methodological tool intended to structure reasoning, not a guarantee independent of usage context or operational environment.

Entrechat’s architecture is not based on the assumption that software can substitute for field investigation, human observation, or analysis of real usage contexts. Social, organizational, and operational factors remain determinant in the effective evaluation of communication situations.

The system operates within existing legal, institutional, and operational frameworks specific to communication technologies. It neither seeks to deny them, nor to oppose them, nor to formulate a global response to these frameworks.

The architectural role assigned to software has been limited. Entrechat restricts itself to communication, session establishment, and cryptographic protection, without integrating qualification, regulation, or automated decision mechanisms regarding usage.

This choice aims to maintain a clear separation between system responsibility and user responsibility, to preserve the coherence of strictly synchronous operation, and to reduce the exposure surface associated with increased functional complexity.

Architectural omission is thus treated as a design choice. The absence of certain features is not presented as a deficiency, but as a means to limit operational exposure, reduce implicit states, and restrict the reasoning space required for system analysis.

4 Operational Exposure Model

The system is designed to operate in environments where capabilities for observing, measuring, delaying, or disrupting network communications may be present.

No assumptions are made regarding:

- the identity of entities possessing such capabilities,
- their scale or scope,
- their intentions or objectives.

Explicitly considered out of scope are:

- full compromise of the user endpoint,
- coercive access to the device or its execution environment,
- data exfiltration resulting from user-side actions.

Reduction of operational exposure relies primarily on explicit architectural omissions: absence of servers, absence of persistent storage, absence of asynchronous signaling, and absence of background network activity outside user-initiated sessions.

Example of a Typical Session (Illustrative)

An Entrechat session begins when two previously identified peers explicitly initiate simultaneous communication. The session is established, messages are exchanged in real time, and communication is terminated by either participant.

At the end of the session:

- no messages are stored infrastructure-side,
- no persistent application state is retained,
- no network activity remains outside user action.

This example is provided for illustrative purposes and constitutes neither a normative model nor an operational guarantee.

5 Cryptographic Model

Entrechat relies on a conventional hybrid cryptographic scheme, adapted to a strictly synchronous peer-to-peer communication model.

- Asymmetric cryptography is used for session establishment [1, 2, 3].
- Message content is protected by authenticated symmetric encryption, limited to the duration of the active session.

Session keys are generated on demand and strictly bounded to ongoing sessions. No persistent inter-session confidentiality nor long-term confidentiality property is claimed.

The architecture excludes:

- any pre-key infrastructure,
- any asynchronous key distribution,
- any deferred message storage,
- any persistent cryptographic state outside the active session.

These exclusions aim to avoid classes of exposure documented in asynchronous messaging systems, notably those related to deferred key management and accumulation of persistent states [7].

6 Architectural Principles

The observed properties of the system result from explicit structural constraints, rather than from protocol or functional sophistication.

The adopted architectural principles are as follows:

- explicit user initiation of communications,
- deterministic and bounded session lifecycles,
- absence of implicit or autonomous network activity,
- absence of background connectivity outside active sessions,
- strictly limited metadata generation.

Outside explicitly initiated sessions, the application generates no network communication.

7 What Synchrony Enables

The choice of a strictly synchronous model is not merely a constraint, but enables certain properties that are difficult to achieve in asynchronous architectures.

In particular, strict synchrony enables:

- absence of application traces outside active sessions,
- absence of deferred or orphaned messages,
- absence of ambiguous states related to delivery or reading,
- immediate understanding of the effective exposure surface at the moment of communication.

These properties do not stem from an additional protective mechanism, but directly from the reduction of functional scope and the absence of persistent states.

8 Limits and Non-Objectives

Entrechat is designed within a restricted scope. The following limitations are inherent to its architectural choices and do not constitute system objectives.

The system does not provide:

- availability or service continuity guarantees,
- message delivery mechanisms outside active sessions,
- protection in case of user endpoint compromise,
- universal or unconditional anonymity,
- guarantees beyond correct local execution of the software.

Entrechat constitutes neither an operated service, nor a platform, nor an infrastructure. It implies no contractual relationship and creates no responsibility regarding uses performed by third parties.

9 Residual Exposures

Identified residual exposures result directly from the system's constraints and architectural choices. They include in particular:

- availability constraints inherent to a strictly synchronous communication model,
- limitations associated with low-latency anonymity when using the Tor network [4],
- loss of local confidentiality in case of user endpoint compromise.

These exposures are not addressed through mechanisms of complexity or concealment. They are explicitly accepted and bounded by the system's architectural omissions.

10 Conclusion

Entrechat v1 presents a strictly synchronous peer-to-peer messaging architecture, based on explicit architectural choices and a deliberately restricted scope.

The system prioritizes structural clarity, deterministic interactions, and control of operational exposure, rather than functional exhaustiveness or protocol sophistication.

Entrechat should be understood as an intentional engineering artifact: specialized, explicitly bounded, and aligned with a precise and assumed usage framework, whose properties result primarily from architectural omission and transparency of retained assumptions.

References

- [1] W. Diffie and M. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, 1976.
- [2] D. J. Bernstein, *Curve25519: New Diffie-Hellman Speed Records*, PKC, 2006.
- [3] A. Langley, M. Hamburg, and S. Turner, *Elliptic Curves for Security*, RFC 7748, IETF, 2016.
- [4] R. Dingledine, N. Mathewson, and P. Syverson, *Tor: The Second-Generation Onion Router*, USENIX Security Symposium, 2004.
- [5] S. Barman et al., *All the Numbers Are US: Large-Scale Abuse of Contact Discovery*, NDSS, 2021.
- [6] A. Greschbach et al., *Measuring Metadata Exposure in Modern Messaging Systems*, USENIX WOOT, 2024.
- [7] J. Müller et al., *Prekey Pogo: Attacking Asynchronous Key Distribution*, IEEE Symposium on Security and Privacy, 2025.
- [8] M. Backes et al., *Careless Whisper: Delivery Receipts and User Activity Leakage*, arXiv:2411.11194, 2024.