Journal of Secure Digital Technologies
Volume 1 (№2) September 2026

ISSN (E): 3136-9340
Sertificate № KZ19VPY00128988, date of issue: 25.12.2025

The journal is published 4 times a year
(March, June, September, December)

EDITORIAL TEAM
Editor-in-Chief
Dr. Leila Rzayeva, Association prof., PhD, Digital government support center (Kazakhstan)

Deputy Editor-in-Chief
Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

Executive Secretary
Dr. Abdenova Zhuldyz, PhD, Digital government support center (Kazakhstan)

EDITORIAL BOARD MEMBERS (KAZAKHSTAN)
Seitkulov E.N. – Candidate of Physical and Mathematical Sciences, Professor Computer Science, Computer Engineering and Control
Abdenova G.A. – PhD in Automation and Control, Candidate of Technical Sciences, Professor L.N. Gumilyov Eurasian National University
Atanov S.K. – Doctor of Technical Sciences, Professor.
Sergazin G.K. – PhD, Associate Professor Computer Science, Computer Engineering and Control
Satybaldina D.Zh. – Candidate of Physical and Mathematical Sciences, Professor Director, Research Institute of Information Security and Cryptology, L.N. Gumilyov Eurasian National University

INTERNATIONAL EDITORIAL BOARD MEMBERS
Ibraheem Shayea – PhD in Electrical and Electronic Engineering Istanbul Technical University (Turkey)
Yadigar Nasiboglu Imamverdiyev – DSc in Technical Sciences, Associate Professor Information Security, Computer Science and Computer Technology (Azerbaijan)
Erdal Irmak – PhD in Electrical and Electronics Engineering, Professor Gazi University (Turkey)
Korhan Kayisli – PhD in Electrical Electronics Engineering, Associate Professor Gazi University (Turkey)
Yonghao Wang – PhD in Electronic Engineering, Associate Professor Queen Mary, University of London (UK)
Amir Mosavi – PhD in Applied Informatics, Professor of AI Obuda University (Hungary)
Gerald Feldman – PhD in General Computing, Associate Professor Birmingham City University (UK)
Mohamed Ragab – PhD in Computer Science, Assistant Professor Birmingham City University (UK)
Seref Sagiroglu – PhD in System Engineering, Professor Gazi University (Turkey)

MANAGING EDITOR
Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

TECHNICAL EDITOR
Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

CONTENTS

Computational Assessment of a Reconstructed Fuzzy Model for Categorizing Critical Digital Objects 3

Cyber Incident Damage Assessment for Public Digital Systems and Critical Infrastructure 27

Verifiable but Not Correct: Blockchain Trust Boundaries for Autonomous Agents 61

Calibrated Digital Twins and Geometry-based Perception for Automated Driving in Heavy Rain and Snowfall: Evidence From Research Testing in China and Implications for Kazakhstan. 78

Personal Data Protection in Kazakhstan's AI Enabled Public Services 99

Asking Another Jurisdiction What It Holds Without Telling It What You Hold: Private Artifact Matching with a Ledger-Anchored Transcript 121

On-Chain and Off-Chain Data Placement in Hybrid Blockchain Health Systems 133

An Integrated Project Management Maturity Model for Digital Qazaqstan Initiatives: Embedding Cybersecurity, AI Governance and Forward-Looking Cyber-Risk Requirements into ST RK ISO 21502-2022 148

Privacy Risks and Data Protection Readiness in AI Marketing in Kazakhstan 173

Bakytzhan Zhumadilla*
PhD student, Department of Informatics, Information and Communication Technologies
zhumadilla_bakytzhan@korkyt.kz, orcid.org/0009-0005-0976-6250
Korkyt Ata Kyzylorda University, Kazakhstan
Nurbek Konyrbaev
PhD, Department of Computer Science
n.konyrbaev@mail.ru, https://orcid.org/0000-0002-8788-4149
Korkyt Ata Kyzylorda University, Kazakhstan
Gul Jussupova
Candidate of Technical Sciences, Department of Digitalization and Public Services
g.jussupova@sci.gov.kz, https://orcid.org/0009-0009-7205-3666
Ministry of Science and Higher Education of the Republic of Kazakhstan, Kazakhstan

ASKING ANOTHER JURISDICTION WHAT IT HOLDS WITHOUT TELLING IT WHAT YOU HOLD: PRIVATE ARTIFACT MATCHING WITH A LEDGER-ANCHORED TRANSCRIPT

Abstract: Two authorities investigating related cases would both benefit from knowing which digital artifacts they hold in common. Neither can afford the usual way of finding out. Sending a list of hashes discloses the whole holding, including artifacts belonging to unrelated subjects, to a partner who may be under no obligation to discard it; and once sent, neither side can prove afterwards what was asked or what was answered. We apply Diffie-Hellman private set intersection over secp256k1 to this problem and add what the forensic setting actually needs, which is an auditable record: the Merkle roots of both blinded transcripts and a digest of the result are committed as a salted field vector and anchored on a permissioned ledger, so the exchange can later be shown to have happened exactly as claimed, with the artifact digests themselves never disclosed. We evaluate on 62,973 real artifacts recovered from a FAT32 medium by The Sleuth Kit and foremost. In a two-device scenario, 32,988 digests against 24,154 with a true overlap of 64, the protocol returns the intersection exactly, no false positives and no false negatives, in 8.4 s and 3.60 MiB of traffic, which is 2.06 times what a plain digest exchange would cost and discloses nothing beyond the intersection. Cost is linear: 66 bytes and 0.24 ms per item held. Anchoring the transcript adds 632 bytes of ledger state and 13 ms, and the disclosable record is 2,166 bytes and verifies in 0.96 ms. We also measure a protocol-hygiene failure that is easy to ship: reusing the blinding exponent across queries makes successive transcripts identical, letting a counterparty track a holding over time, where fresh exponents give zero overlap.

Keywords: *digital forensics; private set intersection; cross-border evidence exchange; permissioned ledger; transcript attestation; secp256k1; mutual legal assistance*

Запрос к другой юрисдикции о её данных без раскрытия собственных: конфиденциальное сопоставление цифровых артефактов с закреплённым в реестре транскриптом обмена

Аннотация: Двум ведомствам, расследующим связанные дела, было бы полезно знать, какие цифровые артефакты есть у них обоих. Ни одно из них не может позволить себе обычный способ это выяснить. Передача списка хеш-значений раскрывает весь массив данных, включая артефакты, относящиеся к посторонним лицам, партнёру, который может быть не обязан его уничтожить; а после передачи ни одна из сторон не может впоследствии доказать, что именно запрашивалось и что было получено в ответ. Мы применяем протокол конфиденциального пересечения множеств (PSI) на основе Диффи-Хеллмана над кривой secp256k1 и добавляем то, что действительно необходимо в криминалистике, а именно проверяемую запись: корни Меркла обоих ослеплённых транскриптов и дайджест результата фиксируются как вектор полей с солью и закрепляются в разрешённом (permissioned) распределённом реестре, так что впоследствии можно показать, что обмен состоялся именно так, как заявлено, при этом сами дайджесты артефактов никогда не раскрываются. Оценка выполнена на 62 973 реальных артефактах, восстановленных с носителя FAT32 средствами The Sleuth Kit и foremost. В сценарии с двумя устройствами (32 988 дайджестов против 24 154 при истинном пересечении 64) протокол возвращает пересечение точно, без ложноположительных и ложноотрицательных результатов, за 8,4 с и 3,60 МиБ трафика, что в 2,06 раза больше стоимости простого обмена дайджестами, при этом сверх пересечения ничего не раскрывается. Стоимость линейна: 66 байт и 0,24 мс на каждый элемент. Закрепление транскрипта добавляет 632 байта состояния реестра и 13 мс, а раскрываемая запись занимает 2 166 байт и проверяется за 0,96 мс. Мы также измеряем ошибку гигиены протокола, которую легко допустить на практике: повторное использование ослепляющей экспоненты в разных запросах делает последовательные транскрипты идентичными и позволяет контрагенту отслеживать массив данных во времени, тогда как при свежих экспонентах пересечение транскриптов равно нулю.

Ключевые слова: *цифровая криминалистика; конфиденциальное пересечение множеств; трансграничный обмен доказательствами; разрешённый распределённый реестр; аттестация транскрипта; secp256k1; взаимная правовая помощь*

Өз деректеріңді ашпай-ақ басқа юрисдикциядан оның деректері туралы сұрау: тізілімде бекітілген алмасу транскриптімен цифрлық артефактілерді құпия салыстыру

Аңдатпа: Өзара байланысты істерді тергейтін екі ведомство үшін қандай цифрлық артефактілер екеуінде де бар екенін білу пайдалы болар еді. Бірақ олардың ешқайсысы мұны анықтаудың әдеттегі тәсілін қолдана алмайды. Хеш-мәндер тізімін жіберу бүкіл деректер жиынын, соның ішінде бөгде тұлғаларға тиесілі артефактілерді, оны жоюға міндетті болмауы мүмкін серіктеске ашады; ал жіберілгеннен кейін тараптардың ешқайсысы не сұралғанын және не жауап берілгенін кейін дәлелдей алмайды. Біз secp256k1 қисығы үстінде Диффи-Хеллман негізіндегі жиындардың құпия қиылысуы (PSI) хаттамасын қолданамыз және криминалистикаға шын мәнінде қажет нәрсені, яғни тексерілетін жазбаны қосамыз: екі соқырландырылған транскриптің Меркл түбірлері мен нәтиже дайджесті тұздалған өрістер векторы ретінде бекітіліп, рұқсат етілген (permissioned) таратылған тізілімге жазылады, сондықтан кейін алмасудың дәл мәлімделгендей өткенін көрсетуге болады, ал артефактілердің дайджесттері ешқашан ашылмайды. Бағалау The Sleuth Kit және foremost құралдарымен FAT32 тасымалдағышынан қалпына келтірілген 62 973 нақты артефактіде жүргізілді. Екі құрылғы сценарийінде (шынайы қиылысуы 64 болатын 32 988 және 24 154 дайджест) хаттама қиылысуды дәл, жалған оң және жалған теріс нәтижелерсіз, 8,4 с ішінде және 3,60 МиБ трафикпен қайтарады; бұл қарапайым дайджест алмасуынан 2,06 есе көп, бірақ қиылысудан тыс ештеңе ашылмайды. Шығын сызықтық: әр элементке 66 байт және 0,24 мс. Транскриптті бекіту тізілім күйіне 632 байт пен 13 мс қосады, ал ашуға болатын жазба 2 166 байтты құрайды және 0,96 мс ішінде тексеріледі. Сондай-ақ біз іс жүзінде оңай жіберілетін хаттама гигиенасы қатесін өлшейміз: соқырландыру дәреже көрсеткішін сұраулар арасында қайта пайдалану бірізді транскрипттерді бірдей етеді және контрагентке деректер жиынын уақыт бойынша бақылауға мүмкіндік береді, ал әр сұрауға жаңа көрсеткіш қолданылғанда транскрипттердің қиылысуы нөлге тең болады.

Түйін сөздер: *цифрлық криминалистика; жиындардың құпия қиылысуы; дәлелдемелермен трансшекаралық алмасу; рұқсат етілген таратылған тізілім; транскриптті аттестаттау; secp256k1; өзара құқықтық көмек*

1. Introduction

Investigations overlap more often than they are allowed to admit. Two agencies, sometimes in two countries, each hold a seized medium, and the question of whether the same file appears on both is frequently the question that would join two cases into one. The obvious way to find out is to send a list of hashes. That is also the way that cannot be defended: a hash list is the whole holding, it covers artifacts belonging to people who are not suspects in the receiving jurisdiction, and the receiving authority is usually under no enforceable obligation to forget it.

The problem has a known cryptographic shape. Private set intersection lets two parties learn the common elements of their sets and nothing else, and the literature is mature [1], [2], [3], [4]. What the forensic setting adds is a requirement the cryptography does not

address. An evidence exchange is an event in a case file. Months later somebody will ask what was requested, what was returned, and whether either side changed its story; and answering that must not require re-disclosing the inputs. A protocol transcript that only the two parties hold does not settle such a dispute, because each of them is the party being doubted.

This paper puts the two halves together. We use Diffie-Hellman PSI over secp256k1, which is the least exotic construction that meets the requirement, and commit the exchange: the Merkle roots of both blinded transcripts and a digest of the agreed result are written as a salted field vector and anchored on a permissioned ledger. Neither side can afterwards claim to have sent a different set, or to have received a different answer, and the anchored material discloses nothing, because every field commitment carries 128 bits of salt.

Everything reported here was executed. The corpus is 62,973 artifacts recovered from a real FAT32 medium with The Sleuth Kit and foremost; the ledger is a three-node etcd Raft cluster with four endorsing processes signing over TCP; the curve operations are libsecp256k1. Intersections are checked against ground truth exactly rather than sampled.

The contributions are as follows.

- An exchange protocol that combines PSI with a ledger-anchored transcript, so that a cross-border artifact query is both privacy-preserving and, afterwards, non-repudiable by either side.
- A measurement on real forensic artifacts rather than synthetic sets: two scenarios drawn from one medium, with intersections recovered exactly at 0.24 ms and 66 bytes per item held.
- A quantified comparison against the plain digest exchange that current practice uses, showing the privacy costs 2.06 times the traffic and roughly a quarter of a millisecond per item.
- A measured protocol-hygiene result: reusing the blinding exponent across queries yields transcripts that are byte-identical, which converts a one-off private query into a persistent tracking channel.

1.1 How Artifact Matching Is Done Now

Operational practice is hash sharing. Reference sets such as the NSRL [5] distribute digests of known files, and victim-identification programmes circulate hash lists of known illegal material so that agencies can recognise what they have already seen. The model works because those lists are meant to be public or semi-public. It does not transfer to case holdings, which are neither, and where the asymmetry is the point: an agency may be willing to reveal that it holds one specific artifact while being unwilling, or legally unable, to reveal the other thirty thousand.

1.2 Private Set Intersection

The Diffie-Hellman construction goes back to Meadows [1] and was developed for practical use by Huberman et al. [2] and De Cristofaro and Tsudik [3]. Its cost is two exponentiations per element per party and one group element per element on the wire, which

is more than modern oblivious-transfer constructions [4], [6] need in raw throughput, but it is round-efficient, needs no correlated randomness, and is short enough to implement without a framework. That last property matters when the code has to be inspected by people who will not accept a black box in an evidential path. Circuit-based and OPRF-based protocols scale better for very large sets and would be the right choice above a few million items; the sets in a single case are usually far smaller than that.

1.3 Ledgers in Forensics

Blockchain work in this area has concentrated on the chain of custody [7], [8], [9], and more recently on attesting analytical output. The exchange itself has received little attention, although it is exactly the kind of bilateral act where both parties have an incentive to remember it differently. Our attestation follows the transparency-log tradition [10], [11] and reuses salted per-field commitments so that the record of an exchange can be disclosed without the exchange's content.

1.4 Positioning

Table 1 sets out the alternatives against the two properties that matter here: what the counterparty learns, and whether the exchange can be reconstructed afterwards by someone who trusts neither party.

Table 1. Ways two authorities can compare holdings

Method	What the partner learns	Traffic per item	Exact result	Auditable later
Send digest list	the entire holding	32 B	yes	no
Bloom filter exchange	membership, with false positives	< 32 B	no	no
OT-based PSI [4], [6]	the intersection only	varies	yes	no
DH-PSI [1], [3]	the intersection only	66 B	yes	no
This work	the intersection only	66 B	yes	yes

2. Methods and Materials

2.1 Problem Setting

Two authorities, A and B, hold sets X and Y of artifact digests taken from media they have lawfully seized. They wish to learn X and Y's intersection. A learns which of its own artifacts B also holds; B learns the same about its own. Neither learns anything about the other's non-matching elements, and neither learns the size of the other's set beyond what the message volume reveals, which we treat as public.

2.2 Adversary Model

C1, the honest-but-curious partner. Follows the protocol and then tries to recover the counterparty's non-matching artifacts from what it received. Against a passive partner the security of the construction rests on the decisional Diffie-Hellman assumption in the group; a blinded point reveals nothing about its preimage without the exponent.

C2, the repeat querier. Runs the protocol against the same partner several times over months, hoping to correlate the transcripts and build a picture of how the partner's holdings change. Section 3.4 shows this attack succeeds completely against an implementation that reuses its exponent and fails completely against one that does not.

C3, the revisionist. After the exchange, one side asserts that it asked a different question, or that the answer was different, typically because the answer has become inconvenient. This is the adversary the cryptography does not address and the ledger does.

Out of scope: a malicious party that deviates from the protocol to extract more than the intersection, for example by inflating its input set with guesses. Malicious-model PSI exists and is more expensive; for the inter-agency setting, where both sides are identified and accountable, we take the semi-honest model with an anchored transcript to be the proportionate design, and say so rather than claiming more.

2.3 Protocol Construction

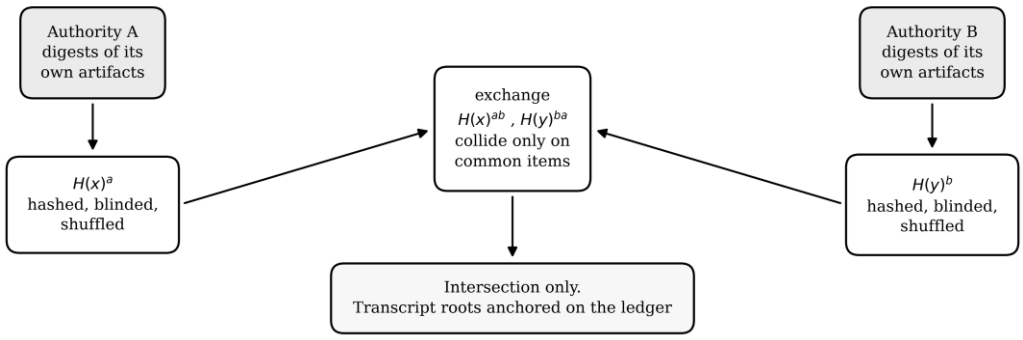


Fig. 1. The exchange. Each authority blinds its own digests with a single-use exponent, both sides re-blind what they receive, and only elements held by both collide. The transcript roots and the result digest are anchored.

Fig. 1 shows the exchange. Each side maps its digests onto curve points and raises them to a secret exponent drawn fresh for this query:

A sends { H(x)^a : x in X }, shuffled (1)

B sends { H(y)^b : y in Y }, shuffled (2)

Each side then raises what it received to its own exponent, producing H(x)^ab and H(y)^ba. Because the group is commutative, an element present in both sets yields the same point on both sides, and an element present in only one yields a point the other party cannot

invert. Matching the two doubly blinded sets gives the intersection, and the shuffle before transmission means position discloses nothing about input order.

Hashing to the curve is by try-and-increment on the compressed encoding, which is not constant time. That is acceptable because the values hashed are the party's own digests, not secrets held from itself; the quantity that must not leak is the exponent, and scalar multiplication is handled by libsecp256k1.

2.4 What the Transcript Commits

The exchange is summarised by four Merkle roots, one over each of the two blinded sets and each of the two doubly blinded sets, together with a digest of the agreed intersection. These are placed in a fixed field vector and each field is committed separately:

$$c = H(0x02 \,\|\, s \,\|\, n \,\|\, enc(v)),\ \ s \leftarrow \{0,1\}^{128} \tag{3}$$

The Merkle root of the field commitments is the exchange digest, and it is that root, not the exchange, which reaches the ledger. A party wanting to establish later what was asked releases the salts for the fields it chooses to open; the rest stay opaque. Because the wire values are already blinded with an exponent that is discarded after the query, opening the transcript roots discloses no artifact digest even to a party that later obtains them.

2.5 Corpus and Scenarios

The corpus is a real medium. A 3.5 GB FAT32 volume was populated with files from 280 distribution packages, 30 percent were deleted with mtools, and the volume was refilled to capacity with further real files, so that clusters were genuinely reused. Artifacts were then taken from it with libtsk and foremost, giving 62,973 artifacts and 57,080 distinct digests. Two scenarios partition those digests, and in both the true intersection arises from duplication that is really present in the corpus rather than being planted.

- Two-device. Authority A holds the digests of the original population, 32,988 of them; authority B holds those of the material written later, 24,154. The true overlap is 64 digests, which is the realistic case: two unrelated devices sharing a handful of common files.
- Even-odd. The artifacts are split by index, giving 29,310 against 29,377 with a true overlap of 1,607. This is the high-overlap operating point, and it tests whether cost depends on how much the sets have in common.

2.6 Experimental Setup

Measurements were taken on two allocated cores under CPython 3.11 with libsecp256k1 through coincurve, single-threaded. A scalar multiplication costs 49 microseconds on this hardware, which is the unit of account for everything that follows. The ledger is a deployed three-node etcd 3.4.30 Raft cluster with four endorsing processes holding Ed25519 keys and answering over TCP sockets.

3. Results

3.1 Correctness and Cost

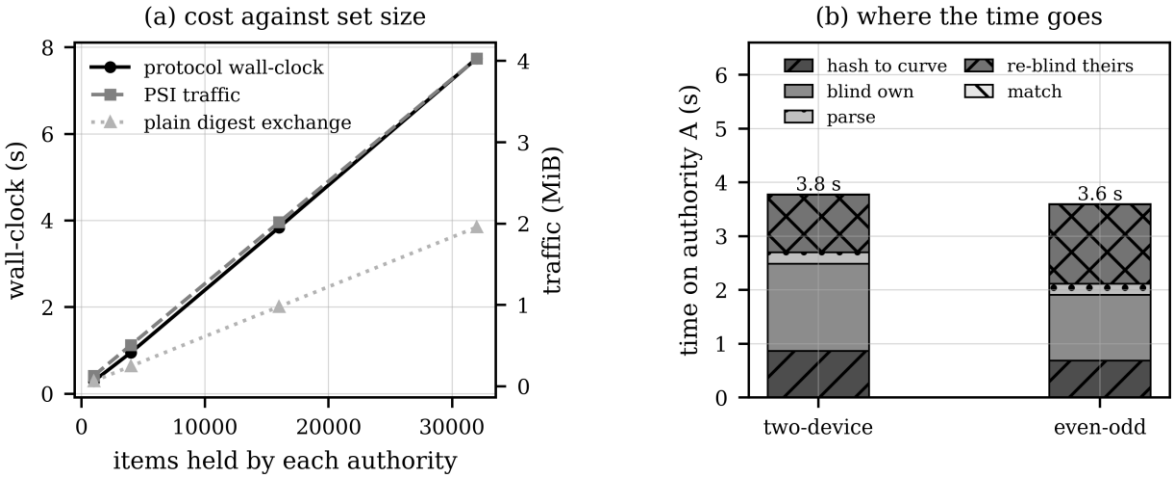


Fig. 2. (a) Wall-clock and traffic against set size, with the plain digest exchange for comparison. (b) Where authority A's time goes in each scenario.

Both scenarios returned the intersection exactly: 64 of 64 and 1,607 of 1,607, with no false positives and no false negatives. Correctness here is not a statistical claim, since the protocol is exact and the ground truth is set equality on digests; the experiment confirms that the implementation realises the protocol. Table 2 gives the costs. The two-device query took 8.4 s and moved 3.60 MiB; the even-odd query, with twenty-five times the overlap, took 7.4 s and moved 3.69 MiB. Cost tracks set size and is indifferent to how much the sets share, which is the privacy property showing up in the performance numbers.

Table 2. Both scenarios, executed on digests of real recovered artifacts

Scenario	A items	B items	True overlap	Found	Wall-clock	Traffic
Two-device	32,988	24,154	64	64 exact	8.4 s	3.60 MiB
Even-odd	29,310	29,377	1,607	1,607 exact	7.4 s	3.69 MiB

Fig. 2(b) breaks down where authority A's time goes in the two-device scenario: 0.87 s hashing its own digests onto the curve, 1.62 s blinding them, 0.21 s parsing what arrived, 1.07 s re-blinding it and 6 ms matching. Two thirds is scalar multiplication, as expected, and the matching step that people intuitively expect to dominate is noise.

3.2 Against the Plain Digest Exchange

The comparison that matters operationally is with what agencies do now. Sending raw digests costs 32 bytes per item once; the protocol costs 66 bytes per item, being a 33-byte compressed point in each of two rounds. The privacy therefore costs 2.06 times the

traffic in both scenarios and about 0.24 ms of computation per item held. On a 30,000-item holding that is under four megabytes and under ten seconds, against a plain exchange that would disclose 30,000 artifacts to obtain 64 answers. Put that way the trade is not close.

3.3 Scaling

Table 3 repeats the protocol on random subsets. Wall-clock rises from 0.29 s at 1,000 items per side to 7.74 s at 32,000, and traffic from 0.13 MiB to 4.03 MiB, both linear in set size to within measurement noise, and every run returned the exact intersection. Linearity is the expected behaviour of the construction and the measurement is a check that no accidental quadratic step crept into the implementation, which is a common way for a matching phase to go wrong.

Table 3. Scaling on random subsets. Every run returned the exact intersection

Items per side	Wall-clock	Traffic	Bytes per item	Plain digest exchange
1,000	0.29 s	0.13 MiB	66	0.06 MiB
4,000	0.94 s	0.50 MiB	66	0.24 MiB
16,000	3.83 s	2.01 MiB	66	0.98 MiB
32,000	7.74 s	4.03 MiB	66	1.95 MiB

3.4 Exponent Reuse

A working implementation can be correct and still leak, and this is the way we expect it to happen in practice. If a party keeps its blinding exponent between queries, its blinded set is a deterministic function of its holdings, so two transcripts are directly comparable. We measured it: with a fresh exponent per query, two runs over the same 3,000 items share none of their wire values, an overlap of 0.000; with the exponent reused, the overlap is 1.000 and every unchanged artifact is recognisable across the two runs. The intersections agree in both cases, so nothing in the output would alert an operator. A counterparty that keeps old transcripts thereby obtains a persistent fingerprint of the other's holdings and can watch it change, which is precisely the surveillance the protocol was adopted to prevent. Deriving the exponent freshly per query costs one random scalar.

3.5 Anchoring the Exchange

Committing the transcript and anchoring it on the Raft cluster costs 632 bytes of ledger state and 13.4 ms in the two-device scenario, against 3.60 MiB and 8.4 s for the exchange it attests: four ten-thousandths of the traffic. The disclosable record, releasing 12 of its 18 fields, is 2,166 bytes and verifies in 0.96 ms including three real Ed25519 signature checks. Table 4 collects these. The practical reading is that there is no cost argument against attesting every exchange.

Table 4. Cost of attesting the exchange, and the effect of reusing the blinding exponent

Quantity	Two-device	Even-odd	Note
Ledger state per exchange	632 B	632 B	constant
Anchor latency	13.4 ms	6.5 ms	real Raft commit
Disclosable record	2,166 B	2,166 B	12 of 18 fields
Record verification	0.96 ms	0.82 ms	3 Ed25519 checks
Wire overlap, fresh exponent	0.000	0.000	runs unlinkable
Wire overlap, reused exponent	1.000	1.000	runs linkable

4. Discussion

4.1 Limitations

Three. The security model is semi-honest: a party that inflates its input set with guessed digests learns whether those guesses are held, and this construction does not prevent it, though the anchored transcript does record the set size that was submitted. The cluster runs on loopback, so the anchoring latency is LAN-class, while the byte counts do not depend on topology. And the corpus, while genuinely recovered from a real medium by real tools, is not a seized device; the overlap between our two scenarios arises from duplicate files in a software distribution rather than from the behaviour of a suspect, so the overlap sizes should be read as operating points and not as estimates of what real cases share.

4.2 What the Anchor Is For

It is worth being precise about the division of work. The cryptography protects the inputs during the exchange. The anchor protects the account of the exchange afterwards, and only that: it establishes that a query with these transcript roots and this result digest completed no later than its epoch was anchored. It says nothing about whether the inputs were honestly assembled. Presenting it as more would repeat the error the custody literature has already made once.

4.3 Fitting Mutual Legal Assistance

Assistance under the Budapest Convention and its second additional protocol [12] is slow partly because the request itself discloses. An authority must often explain what it is looking for in order to ask, and that explanation reaches a foreign administration. A private query inverts the order: the parties establish that there is something to talk about, and only then does the formal process begin, now with a narrow and well-founded scope. The anchored transcript gives the later formal step something to point at.

4.4 Data Protection

Nothing identifiable is written to the ledger. Ledger state consists of epoch roots, sizes, linkage and signatures, and every field commitment is salted with 128 bits of entropy, so the anchored record is not personal data recoverable by anyone, which is the property that makes erasure obligations under the GDPR [13] satisfiable off-chain. The blinded transcripts themselves are not retained by the protocol; if a party keeps them, the exponent that produced them is gone.

5. Conclusion

Two agencies can find out what they hold in common without either handing over its holdings, and the cost of doing so on real forensic artifacts is about twice the traffic of the disclosure-everything method practised today and a quarter of a millisecond per item. Both of our scenarios returned the intersection exactly. The part that the cryptographic literature leaves out, and that a case file needs, is the record: anchoring the transcript costs 632 bytes and 13 ms, four ten-thousandths of the exchange it attests, and leaves a claim about what was asked and answered that neither party can later revise. The failure mode to watch is not the mathematics but the housekeeping. An implementation that reuses its blinding exponent produces identical transcripts across queries and turns a private question into a standing one, and nothing in the protocol's output reveals that this has happened.

Author Contributions

Conceptualization – B.Zh; Methodology – N.K. and G.J.; Software – B.Zh; Validation – B.Zh; Writing - original draft – B.Zh; Writing - review and editing N.K. and G.J.; Supervision – B.Zh.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1]. Meadows, C. (1986). A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party. In *Proceedings of the IEEE Symposium on Security and Privacy* (pp. 134–137). IEEE.

[2]. Huberman, B. A., Franklin, M., & Hogg, T. (1999). Enhancing privacy and trust in electronic communities. In *Proceedings of the 1st ACM Conference on Electronic Commerce* (pp. 78–86). ACM.

[3]. De Cristofaro, E., & Tsudik, G. (2010). Practical private set intersection protocols with linear complexity. In *Financial cryptography and data security* (LNCS, Vol. 6052, pp. 143–159). Springer.

[4]. Pinkas, B., Schneider, T., & Zohner, M. (2014). Faster private set intersection based on OT extension. In *Proceedings of the 23rd USENIX Security Symposium* (pp. 797–812). USENIX Association.

[5]. National Institute of Standards and Technology. (2024). *National Software Reference Library and the Reference Data Set*.

[6]. Kolesnikov, V., Kumaresan, R., Rosulek, M., & Trieu, N. (2016). Efficient batched oblivious PRF with applications to private set intersection. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security* (pp. 818–829). ACM.

[7]. Lone, M. M., & Mir, R. N. (2019). Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer. *Digital Investigation, 28*, 44–55.

[8]. Bonomi, S., Casini, M., & Ciccotelli, C. (2018). *B-CoC: A blockchain-based chain of custody for evidences management in digital forensics* (arXiv:1807.10359). arXiv.

[9]. Dasaklis, T. K., Casino, F., & Patsakis, C. (2021). SoK: Blockchain solutions for forensics. In *Technology development for security practitioners* (pp. 21–40). Springer.

[10]. Merkle, R. C. (1988). A digital signature based on a conventional encryption function. In *Advances in Cryptology - CRYPTO 1987* (LNCS, Vol. 293, pp. 369–378). Springer.

[11]. Laurie, B., Langley, A., & Kasper, E. (2013). *Certificate transparency* (RFC 6962). Internet Engineering Task Force.

[12]. Council of Europe. (2001). *Convention on Cybercrime* (ETS No. 185), and Second Additional Protocol (CETS No. 224, 2022). Council of Europe.

[13]. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation). (2016). *Official Journal of the European Union, L 119*, 1–88.

[14]. Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *Proceedings of the USENIX Annual Technical Conference* (pp. 305–319). USENIX Association.

[15]. Faz-Hernandez, A., Scott, S., Sullivan, N., Wahby, R. S., & Wood, C. A. (2023). *Hashing to elliptic curves* (RFC 9380). Internet Engineering Task Force.

[16]. Carrier, B. (2005). *File system forensic analysis*. Addison-Wesley.

[17]. Casey, E. (2011). *Digital evidence and computer crime: Forensic science, computers, and the Internet* (3rd ed.). Academic Press.

[18]. International Organization for Standardization. (2012). *Information technology - Security techniques - Guidelines for identification, collection, acquisition and preservation of digital evidence* (ISO/IEC Standard No. 27037:2012).