Privacy-Friendly Checking of Remote Token Blacklists

Roel Peeters, Andreas Pashalidis

Privacy-Friendly Checking of Remote Token Blacklists

Roel Peeters and Andreas Pashalidis

KU LEUVEN, ESAT/SCD - COSIC & iMinds
Kasteelpark Arenberg 10 bus 2446, 3001 HEVERLEE, BELGIUM
firstname.lastname@esat.kuleuven.be

Abstract. Consulting a remote blacklist as part of verifying a token should not come at the cost of privacy. In particular, the blacklist provider should be unable to identify which tokens are being verified. The contents of the blacklist should also be protected; that is, it should not be possible to learn the contents of the blacklist, for example by querying the blacklist provider a large number of times. This paper defines a range of desirable properties for privacy preserving blacklist checking protocols, and surveys existing technical solutions to this problem. We propose adaptations where appropriate, and provide concrete performance estimates for the use case of checking whether or not a passport has been reported lost or stolen.

1 Introduction

As part of verifying a token it is sometimes necessary to check with a remote authority, in an online fashion, whether or not the token has been blacklisted. 'Transport layer security' (TLS) clients such as browsers, for example, can be configured to query a remote 'online certificate status protocol' (OCSP) server as part of verifying a server certificate. This query contains the serial number of the encountered certificate, and the OCSP server's response indicates whether or not the corresponding certificate has been revoked. Similarly, as part of verifying an official document such as an identity card or a passport, inspection systems sometimes issue a query to remote authorities that maintain document blacklists. This query, too, contains the serial number of the document and the response indicates whether or not the document has been blacklisted.

Currently deployed systems for blacklist checking reveal the identity of the token to the remote authority. This situation is unfortunate, because it undermines the privacy of the token owner. In the OSCP setting, for example, the server gets to know with whom the verifier is about to communicate (namely the owner of the token), and in the official document verification scenario, the remote database authority gets to know whose documents are being verified. Since inspection systems are typically located at well-known locations such as border crossings and airports, this reveals citizens' travel patterns.

A more privacy-friendly approach to revocation checking involves pushing the entire blacklist to every verifier. In this way, verifiers can perform the blacklist check locally, without consulting any remote authority. In fact, 'certificate

revocation lists' (CRLs) follow exactly this approach. However, CRL-like solutions are often unacceptable because the blacklist is itself sensitive and must not be disclosed to verifiers. Otherwise, criminals with illegitimate access to an inspection system, for example, will be able to check, without risking detection, whether or not any stolen passports have already been blacklisted. Currently, Interpol offers itself an online query interface to its 'stolen or lost travel document' (SLTD) database, or may arrange for a copy of the database to be pushed to a national server. The database, is, however, never pushed to individual inspection systems[1].

The remainder of this paper is structured as follows. The next section defines our model of remote blacklist systems and desirable properties. Section 3 describes a simple solution that achieves some of the properties, and Sect. 4 surveys the literature in search of schemes that outperform the simple solution. Section 5 describes our proposal, which is an adaptation of an existing techniques. Finally, Sect. 6 concludes.

2 Model and desirable properties

This section describes our abstract model for blacklist systems. First, we describe our system model. Next, the definitions of the desirable properties are given. Finally, we discuss the adversary model.

2.1 System Model

A remote blacklist system consists of three types of players, namely tokens, verifiers, and a blacklist provider. It is assumed that each token is assigned a unique identifier from the universe of token identifiers $\mathcal{T}$, and is issued to a user. In our system description, we assume that a single verifier exists in the system, and denote it by V.[2] The remote blacklist provider is denoted by BP and has a collection of blacklists $B_1, B_2, \ldots$, with each blacklist $B_v = \{\tau_1, \tau_2, \ldots, \tau_{|B_v|}\} \subseteq \mathcal{T}$ containing $|B_v|$ distinct token identifiers, where v denotes a blacklist's version number. The way in which BP constructs the blacklist collection is outside the scope of this paper. It is assumed that BP and V communicate over a secure channel.

A remote blacklist system also defines two protocols, namely Init and Query. The Init protocol is executed between BP and V, where BP's input is B_v. It is executed at least once, and may be executed regularly in fixed intervals or on demand. Without loss of generality, we assume that, when Init is executed for the vth time, BP's input is B_v (for all $v \in \{1, 2, \ldots\}$).

The Query protocol is executed whenever V wishes to verify a token, and involves all three parties. In some systems, the token has a bidirectional communication interface (e.g. eID documents, smartphones), while in others this

[1] `https://www.interpol.int/Public/FindAndMind/default.asp`

[2] Sections 4.2 and 5 consider the setting of multiple verifiers.

interface is unidirectional (e.g. the machine readable zone of a document, a public key certificate). Tokens with a bidirectional interface may play an active role in the protocol, i.e. react to incoming messages, while tokens with a unidirectional channel are typically passive. That is, V simply reads information from the token (e.g. its identifier τ) and uses it during the protocol. BP's input to the protocol is B_v for a given version v, and V's input is τ, the identifier of the token. At the end of the protocol execution, V learns whether or not $\tau \in B_v$.

2.2 Desirable properties

A remote blacklist system must enable V to obtain the required information. That is, it must satisfy *correctness*: at the end of a Query protocol execution where BP's input is B_v and V's aim is to check the status of the token with identifier τ, V learns whether or not $\tau \in B_v$. In the following, we list further desirable properties for a remote blacklist system.

- **T-User Privacy.** Given a well-defined subset of token identifiers $T \subseteq \mathcal{T}$, for each Query protocol execution BP learns nothing beyond whether or not $\tau \in T$. Note that T must be fixed before the Query protocol execution starts. Also, if $T = \mathcal{T}$, then the strongest possible privacy notion is achieved as BP essentially learns nothing from Query protocol executions.
- **Weak T-User Privacy.** The definition of T-User Privacy is relaxed such that BP may also learn τ if $\tau \in T$.
- **Marginal Blacklist Hiding** For each Init protocol execution, V learns nothing about $B_1, \ldots, B_v$ beyond the fact that the blacklist is of length at most $|B_v| + \mu_v$ for some well-defined margin $\mu_v \in \mathbb{Z}^*$, and, for each Query protocol execution, V learns nothing about $B_1, \ldots, B_v$ other than (a) whether or not $\tau \in B_v$, and (b) the upper bound for the blacklist size, as described above.
- **Efficiency.** The Init and Query protocols must have computation and communication complexity that scales well in $|B_v|$; in particular, for the Query protocol, a constant complexity is desirable.
- **Token Binding.** We define multiple variants of this property, as follows.
  - **Reactive -** V must be able to convince an independent auditor that it executed the Query protocol only for tokens that were actually involved in these protocol executions.
  - **Weak Proactive -** V must not be able to execute the Query protocol with BP for a token that was not involved in a Query protocol execution.
  - **Strong Proactive -** V must not be able to execute the Query protocol with BP for a token that is not involved in the *current* protocol execution. This property requires the token to be active, i.e. to respond to incoming messages.
- **Online operation.** BP can authorise each incoming query to a specific version of the blacklist. This is achieved if BP's participation is required in the Query protocol. This property prevents a compromised verifier from executing the Query protocol a large number of times without risking detection.

Strong proactive token binding implies weak proactive token binding, but proactive token binding does not imply reactive token binding (and vice versa). Moreover, in some applications, e.g. certificate revocation checking for websites, a zero margin may be acceptable if disclosing the exact value of the size of the B_v is not a problem. In other applications, such as the list of lost or stolen passports, a positive margin is required as the exact value of $|B_v|$ must remain hidden. Otherwise, an adversary that can execute the system's protocols with BP and that recently stole a moderate number of passports can easily deduce the point in time at which these passports appear in the blacklist. ‘

In the use case of passport revocation checks, we expect a realistic blacklist size to contain several million entries. Despite such a lengthy blacklist, and in order to avoid accumulating queues in front of border control guards at a busy airport, a Query protocol execution must complete within a few seconds. Moreover, token binding is particularly desirable in the context of passport inspection, because it contributes to fraud detection and auditing towards data protection compliance. A solution that supports strong proactive token binding cannot, however, fully replace solutions without this property. This is because there will always be passports without a chip or with a broken chip that nevertheless will need to be checked against the blacklist. Interestingly, a scheme that provides strong token binding, or a scheme with $\mathcal{T}$-user privacy where the passport chip plays a critical role, would provide an incentive for citizens *not* to destroy their passport chips for privacy reasons.

2.3 Adversarial Model

The blacklist provider can always refuse to run the query against the blacklist. We assume that BP has an interest to prevent the acceptance of blacklisted tokens and that it will therefore not deny its service, unless it suspects that the verifier is compromised. However, while BP is assumed to be honest in this respect, it may attempt to determine the identities of tokens that are not truly blacklisted, for example by running incoming queries against hidden ‘shadow’ blacklists in parallel to the real blacklist. Such parallel and invisible query evaluations should be prevented. In this sense, we consider an honest-but-curious BP. Note that systems that provide $\mathcal{T}$-user privacy or (weak) B_v-user privacy automatically also provide protection against verifiers that maintain shadow blacklists.

V may also misbehave, for example by providing a token identifier that does not correspond to a token that is currently under inspection. Since V's goal is to verify the legitimacy of tokens, we consider such behaviour to be outside V's interest. V may, however, become compromised and then serve as a vehicle for the adversary to check the status of stolen tokens, or to otherwise disrupt the system by issuing superfluous queries. The different variants of token binding limit the damage caused by compromised verifiers: while reactive token binding enable detection of compromised verifiers, systems that provide proactive token binding ensure that compromised verifiers cannot query BP about tokens that are not present. Moreover, the property of online operation ensures that BP can

keep a record of how many queries are issued by each verifier, and can hence trigger an alarm if some verifier issues an abnormal amount of queries.

3 A simple solution

This section describes a simple solution to the above problem. This solution makes use of a technique called 'oblivious polynomial evaluation', and essentially adapts the protocol proposed by Freedman *et al.* [8] to our setting.

The Init protocol proceeds as follows. First, BP constructs a random polynomial $f(\cdot)$ of degree $|B_v|$, the roots of which correspond to the elements on the blacklist. Using a homomorphic encryption scheme, each coefficient of the polynomial is then encrypted using BP's public key, and the resulting ciphertexts are sent to V. V stores these ciphertexts.

In order to execute the Query protocol for token identifier τ, V obliviously evaluates the encrypted polynomial $f(\cdot)$ at τ. This is done in ciphertext space and is possible due to the homomorphic properties of the encryption scheme. The result is then multiplied with a random number r and is sent to BP. By decrypting the received value, BP obtains the value $rf(\tau)$. If τ is a root of the polynomial, then $rf(\tau) = 0$ and BP learns that $\tau \in B_v$. Otherwise, the decryption yields a random number and hence does not reveal anything about τ beyond the fact that $\tau \notin B_v$. In both cases, BP returns a bit to V that indicates whether or not $\tau \in B_v$.

Table 1 lists the security and efficiency properties achieved by the simple scheme described above. For the efficiency evaluation, it is assumed that, during a Query protocol run, V uses the hashing-to-bins method as described in [8]. We stress that, in this scheme, the complexity of the Init protocol is amortized over a potentially large number of Query protocol executions.

Table 1. Properties fulfilled by simple scheme.

Property	Comment			
T-User privacy	Yes, for $T = B_v$			
Blacklist hiding	Yes (zero-margin), if BP uses a fresh key for every version v.			
Token binding	No			
Online operation	Yes			
Efficiency	Init		Query	
	BP	V	BP	V
(computation)	$\mathcal{O}(\lvert B_v\rvert)$	-	$\mathcal{O}(1)$	$\mathcal{O}(\ln\ln\lvert B_v\rvert)$
(communication)	$\mathcal{O}(\lvert B_v\rvert)$		$\mathcal{O}(\lvert B_v\rvert)$	

Note that the original protocol as described in [8] requires V to return an encryption of $rf(\tau) + \tau$ (rather than an encryption of $rf(\tau)$) to BP; since this enables BP to pinpoint the identity of the blacklisted token if it appears on

the blacklist, this protocol variant only provides weak B_v-user privacy. Moreover, while the original protocol uses Pallier encryption, in the modified version described above it is possible to further improve computation efficiency using exponential ElGamal. This variant of ElGamal does allow decryption of ciphertexts, but one can test if a ciphertext corresponds to a given plaintext, and this suffices in our setting. However, this is only an improvement in the constant factor, the asymptotic complexity remains the same.

This simple scheme suffers from two major shortcomings: $\mathcal{O}(|B_v|)$ elements need to be tranferred for each Query protocol, and the scheme does not provide any form of token binding.

4 Survey

In the literature, systems that simultaneously address user privacy and blacklist hiding are, depending on their exact properties, said to solve the problem of 'private disjointness testing' (PDT), 'private set intersection' (PSI), 'private set intersection cardinality' (PSI-CA), or 'authorised private set intersection' (APSI). Note these problems are more general than ours: they focus on the case where V queries multiple identifiers in a single protocol execution, whereas the Query protocol as defined in Sect. 2.1 requires V's input to be a single identifier.

In most of the works we discuss below, the 'client', not the 'server', obtains the result of the protocol. While at first glance it may appear more natural for V to assume the role of the client in our setting, we sometimes reverse the roles of the two parties such that BP obtains the result instead. This role reversal yields a significant efficiency advantage because multiple Query protocol executions can be performed after a single Init protocol execution. That is, the effort of Init is amortized over a potentially large number of queries. Note that systems that provide (weak) B_v-user privacy, i.e. systems where BP learns the outcome of the Query protocol (and informs V in a subsequent message), are likely to be considered sufficiently privacy-friendly with respect to users, because only a small number of tokens will be blacklisted. It is important, however, to ensure that V is unable to run queries against shadow lists.

4.1 Schemes without proactive token binding

This section surveys related work on protocols that do not support token binding.

Oblivious Polynomial Evaluation We now briefly review certain schemes that build on [8], and show that, while they offer advantages in the generic PSI and PSI-CA setting, in our case where V 's input to the Query protocol is a single element, they essentially degenerate to the simple solution described above.

- Kiayias and Mitrofanova [13] proposed a variant that uses superposed encryption which offers advantages when V's set is large. However, the usage of superposed encryption offers no advantages in our setting and hence can be omitted.

– Hohenberger and Weis [9] propose a variant that is secure against a dishonest V being able to illegitimately convince BP that the intersection is not empty[3]. However, in our setting it is in V's interest to provide the correct input to the protocol, and hence this type of protection is not required. Note that, solutions that provide some form of token binding must provide security against dishonest verifiers; however, token binding is outside the scope of [9].
– The solution of Ye *et al.* [20, 21] is based on Sylvester matrices. For both datasets, the data serves as the roots of a polynomial. From these two polynomials the Sylvester matrix is constructed. The determinant of this matrix indicates whether or not the two sets intersect. The privacy is protected by encrypting the polynomials with an additive homomorphic encryption scheme, such as Paillier's[4]. To avoid that one could also learn the cardinality of the set intersection, the determinant is evaluated in a secure two-party computation. However, since the set of the verifier only contains a single element, this is not an issue and we can use the protocol as being sketched in the intuition section of [20, 21].

In our setting, the asymptotic communication and computation complexity of the above schemes are identical to the complexity of the simple scheme.

OT and PIR-based techniques Schemes that do not use the idea of polynomial evaluation, such as 'private information retrieval' (PIR), 'symmetric PIR' (SPIR), and 'oblivious transfer' (OT) schemes, can also be used for privacy-friendly checking of remote blacklists. PIR schemes enable a client to retrieve some data items from a database server, without the server learning which items are retrieved. SPIR schemes also protect the privacy of the database, in that the client can only learn a single data item whose index is fixed in advance. The difference between SPIR and OT schemes is that, while the former require a communication complexity that is sublinear in the size of the database, the latter do not.

Naor and Pinkas proposed OT protocols with adaptive queries [15]. In these protocols, the client is allowed to learn at most k out of n data items, and while it can adaptively decide which ones to receive, the database server does not learn anything about the client's choices. The authors showed that these protocols can also be used for the client to learn whether or not an element exists in the database. In terms of efficiency, their protocol requires $\mathcal{O}(\log n)$ invocations of an 'one-out-of-two' oblivious transfer protocol [7].

Chor, Gilboa and Naor introduced Private Information Retrieval (PIR) by keywords [2]. Later, Ogata and Kurosawa introduced the more efficient notion of Oblivious Keyword Search (OKS), where the client learns the data items associated with his keyword privately [18]. This is a form of PSI with additional

[3] In [8] this is possible, for example, by returning an encryption of zero to BP.
[4] Since this solution requires that the ciphertext needs to be decrypted, one cannot use the more efficient exponential Elgamal.

data transfer. One of their protocols, namely the one based on RSA blind signatures, is very efficient and proceeds as follows. Initially, the server generates an RSA signature key pair and signs the keywords $w_1, w_2, \ldots$ yielding signatures $K_1, K_2, \ldots$. The server then commits to the each item of content c_i by computing the commitment $E_i = G(w_i||K_i||i) \oplus (0^l||c_i)$, where $G()$ is a pseudo-random function, w_i is the keyword associated with c_i, and l is a security parameter. The server then sends these commitments to the client, which subsequently asks the server for a blind signature on the keyword w of interest. After unbliding the received signature, denoted K, the client computes $G(w||K||i) \oplus E_i$ for every commitment E_i and, if the result has an l-bit prefix of zeroes, then the remaining bits constitute c_i.

De Cristofaro and Tsudik proposed a PSI protocol [4, Fig. 4], essentially removing the data transfer from [18]. As a result this protocol is more efficient than the one proposed by Ogata and Kurosawa, although the asymptotic complexity remains the same. If this scheme is used in our setting, then the communication complexity of the Init and the Query protocols are $\mathcal{O}(|B_v|)$ and $\mathcal{O}(1)$, respectively. The computation complexity for the Query protocol are $\mathcal{O}(|B_v|)$ for the client and $\mathcal{O}(1)$ for the server.

Huang, Evans, and Katz [10] proposed an approach based on garbled circuits, and showed that their approach is typically more efficient faster than De Cristofaro and Tsudik's protocol. De Cristofaro and Tsudik [5, 6], however, optimized their protocol in terms of efficiency which outperform garbled circuits. The identified drawbacks are that the scheme does not provide blacklist hiding across blacklist versions and it is not clear how to achieve proactive token binding, i.e. how to convert it to an APSI system.

Privacy-Preserving Revocation Checking Solis and Tsudik argue that PIR techniques are too heavyweight for the purposes of blacklist checking, and they proposed a simpler alternative that uses the idea of certificate revocation trees [19]. Narisimha, Solis and Tsudik extended this idea to also include certificate revocation lists [16]. Both proposals require the client to query a range of k elements instead of a single element. An advantage of this approach is that there is no initialization phase, so for a small number of queries the communication overhead outperforms traditional CRLs. However, the degree of privacy depends on the size of the range interval; there is a trade-off between privacy and the number of elements to be transmitted. Furthermore, the systems do not provide blacklist hiding in our setting since the client also learns whether or not elements that have *not* been queried are on the blacklist.

4.2 Schemes supporting proactive token binding

Oblivious Signature-Based Envelope (OSBE) schemes enable a server to send a message to a client such that (a) the client obtains the message if and only if it is authorized to do so by a trusted third party, and (b) the server does not learn whether or not the client was given such authorization [14,17]. The trusted

third party authorizes clients by issuing a special digital signature, namely an 'OSBE signature' σ, over another message μ that is known to both the client and the server. OSBE schemes typically require the client to first send a 'blinded' version of σ to the server. The server then combines the received value with μ and adds extra randomness to it such that the server ends up with two values: a response for the client, and a symmetric encryption key that it uses to encrypt the message. The response together with the encrypted message is returned to the client. Due to the construction of the scheme, the client can only recover the symmetric key, and hence decrypt the ciphertext, if σ is a signature over μ.

OSBE schemes can be used to construct a remote blacklist system that, apart from $\mathcal{T}$-user privacy and zero-margin blacklist hiding, supports weak proactive token binding. To this end, it is required that the token issuing authority embeds an OSBE signature into the tokens it issues and that V extracts it at inspection; we assume that the signed message is the identifier of each token. In order to execute the Query protocol, V then simply proceeds according to the OSBE system: first it blinds the signature and sends it to BP. The blacklist provider then derives a response and a key for each entry on B_v, and encrypts a well-known message, e.g. '`lost or stolen' under each key. The resulting responses and ciphertexts are then sent to $\mathsf{V}$ (in a random order) which then uses its knowledge of the OSBE signature in order to derive a key for each received response. If the ciphertext corresponding to any derived key decrypts to '`lost or stolen', then the client concludes that the token has been blacklisted. Note that, using this approach, the client essentially derives $|B_v|$ symmetric keys in order to identify at most a single entry from the (randomized) blacklist.

It is possible to achieve a constant-factor reduction of the communication and computation complexity by replacing the encryptions of '`lost or stolen' with a one-way hash value of each key. This has been suggested in the context of 'privacy-preserving policy-based information transfer' (PPIT) [3] and is also used in other contexts (e.g. for authentication based on a commitment that is hidden behind a hash function [12]). The asymptotic communication and computation complexity of the Query protocol, however, remains $\mathcal{O}(|B_v|)$. We stress that this cost is likely to be prohibitive in the scenario of passport inspection at busy airports. However, adopting this approach would not introduce incompatibility with current standards, since the signature σ can occupy one of the unused fields of the standard ePassport application specified in [11].

Based on a variant of PPIT that uses RSA signatures, De Cristofaro and Tsudik proposed a more efficient APSI protocol [4, Fig. 2]. Figure 1 shows our adaptation of this protocol to the remote blacklist setting. In this figure, the modulus N, the exponent e and the full-domain hash function H_1 constitute the public RSA signature key of the token issuer, σ denotes the token issuer's signature on the token identifier τ, H_2 is a one-way hash function, and g is a generator of the subgroup of quadratic residues modulo N. It is assumed that the parameters N, e, H_1, H_2, g are known to both V and BP.

While the original system in [4, Fig. 2] consists of a single protocol, our adaptation divides it into the Init and the Query protocols, as shown in the figure.

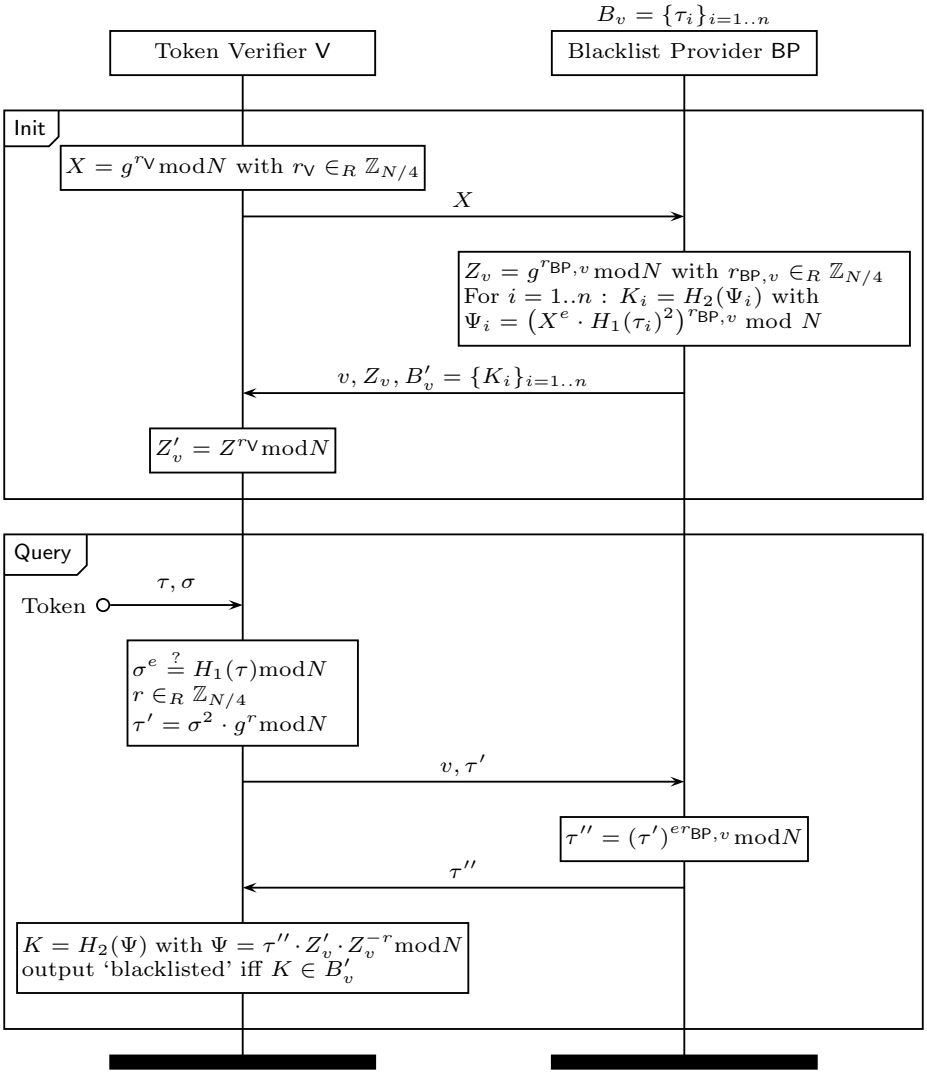

Fig. 1. Scheme from [4, Fig. 2] adapted to the remote blacklist setting

The Init protocol starts with V choosing a random number r_V and sending $X = g^{r_\mathsf{V}}$ to BP. The blacklist provider then chooses its own random number $r_{\mathsf{BP},v}$ and, for each blacklisted token identifier $\tau_i \in B_v$, computes the value $K_i = H_2((X^e \cdot H_1(\tau)^2)^{r_{\mathsf{BP},v}})$ and sends these values along with the version identifier v and the value $Z_v = g^{r_{\mathsf{BP},v}}$ to the verifier. V then computes the value $Z'_v = Z^{r_\mathsf{V}}$ and stores it together with the 'blinded' blacklist $B' = \{K_1, K_2, \dots\}$ and its version number. Note that the complexity of the Init protocol is $\mathcal{O}(|B_v|)$.

The Query proceeds as follows. First, V obtains the values τ and σ from the token. Then it verifies the signature, chooses a random number r from $\mathbb{Z}_{N/4}$, and computes the value $\tau' = \sigma^2 \cdot g^r \mod N$ and sends τ' to the blacklist provider. In order to avoid any ambiguity in the presence of multiple blacklist versions, V also includes v in this message. BP then computes $\tau'' = (\tau')^{er_{\mathsf{BP},v}} \mod N$ and returns τ'' to V. Finally, the verifier computes $K = H_2(\tau'' \cdot Z'_v \cdot Z_v^r) \mod N$ and checks whether or not $K \in B'$. If it is, then it concludes that τ has been blacklisted.

As the original system [4, Fig. 2], our modified scheme above provides zero-margin blacklist hiding, online operation, and weak proactive token binding. As a result of dividing the scheme into two protocols, the complexity of the Query protocol is decoupled from the size of the blacklist; it is merely $\mathcal{O}(1)$. Moreover, the complexity of the Init protocol is amortized over a potentially large number of queries. However, in contrast to the original scheme, the same 'blinded version' of the blacklist B' that BP sends to V during the Init protocol is reused over multiple Query protocol executions. It remains to be shown that this does not introduce security issues.

A shortcoming of the scheme in Fig. 1 is that does not support blacklists that contain tokens issued by multiple issuers. That is, it provides only $\mathcal{T}_{\mathcal{I}}$-user privacy, where $\mathcal{T}_{\mathcal{I}}$ is the set of token identifiers that are signed by a given issuer using a particular signature key. This is because, for each blacklisted token, the blacklist provider must use, in its computations, the parameters of the corresponding issuer signature key; hence, the blacklist provider must be made aware to which 'group' the token that is currently being queried belongs. In the context of passport inspection, this means that it is not possible to hide the nationality of travellers from the blacklist provider. Our proposal in the next section does not suffer from this drawback.

5 Our proposal

While the De Cristofaro and Tsudik's PSI scheme [4, Fig. 4], based on RSA blind signatures, is among the most efficient schemes proposed in the literature, it does not provide any form of token binding. Reactive token binding can nevertheless be achieved simply by requiring (a) the blacklist provider to keep a log of all blind signatures received) and (b) the token verifier to keep a log of all tokens identifiers, their signatures, and the randomness used to blind the signatures. Using these logs, an auditor can identify any superfluous queries that V issued to the blacklist provider, and initiate further investigation, as appropriate.

This section presents our proposal for a privacy-preserving remote blacklist checking. Like the scheme in [4, Fig. 4], it is based on RSA blind signatures. Section 5.1 presents our protocol and Sect. 5.2 compares its complexity to certain other schemes.

5.1 Protocol description

We assume that the token issuer embeds the token identifier τ and its signature σ on τ into the token. In contrast to the requirements of the scheme shown in Fig. 1, we do not require the issuer to use any particular signature scheme. We further assume that the blacklist provider knows σ for every blacklisted token. While this may seem to be an additional burden, it ensures that strict procedures are followed when adding tokens to the blacklist. That is, a token can be added only in cooperation with the issuer or someone that has or had physical access to the token.

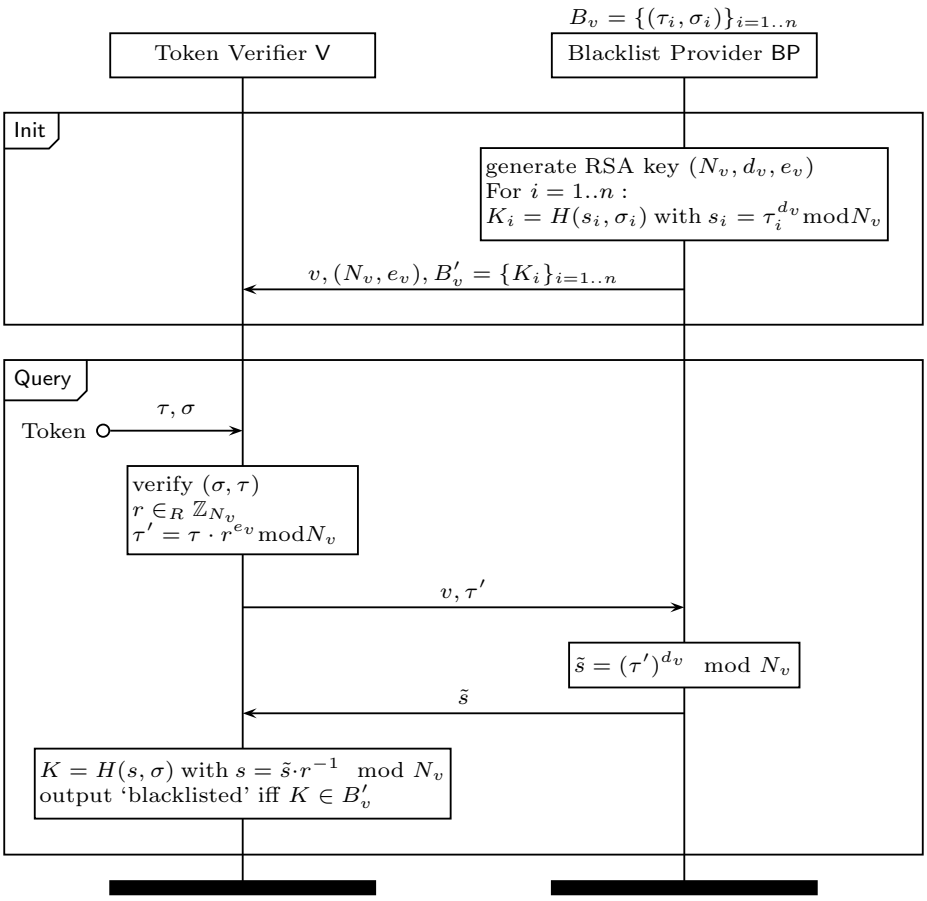

Fig. 2. Privacy-Friendly Checking of Remote Token Blacklists.

Figure 2 shows our proposed scheme. In this figure, H denotes a one-way hash function that V and BP agree on. The Init protocol proceeds as follows. First, BP generates a fresh RSA signature key pair where (N_v, e_v) is the public

verification key and d_v the secret key. Note that this key is used only for the current blacklist version. The blacklist provider then signs every blacklisted token using this key; s_i denotes the signature over the ith entry in the blacklist. Finally, it computes $K_i = H(s_i, \sigma_i)$ where σ_i denotes the token issuer's signature for the corresponding entry, and sends these values, along with its public key (N_v, e_v) and the version number v to V. The verifier stores the received data.

The Query protocol proceeds as follows. First, V obtains τ and σ from the token. It then verifies σ, chooses a blinding factor r from $\mathbb{Z}_{N_v}$ and, using r together with the verification exponent e_v, it randomizes τ to obtain $\tau' = \tau \cdot r^{e_v}$. It then sends τ' together with the version number v (in order to avoid ambiguity) to BP. The blacklist provider then blindly signs the received value and obtains the blind signature $\tilde{s} = (\tau')^{d_v}$ which it sends back to V. Note that $\tilde{s}$ is a blind RSA signature on τ. By removing the blinding factor the verifier obtains $s = \tilde{s} \cdot r^{-1}$ and computes $K = H(s, \sigma)$. If this is identical to any of the values K_i on the blinded blacklist, then V concludes that the token is blacklisted.

Since the issuer's signature σ must be known to V in order to execute the protocol correctly, and since we assume that this signature can be obtained only from the token itself, our proposed scheme provides weak proactive token binding. It also provides $\mathcal{T}$-user privacy, zero-margin blacklist hiding, and online operation. Furthermore our proposed scheme has two advantages over the one discussed in Fig. 1: firstly, the blacklist can contain tokens issued by multiple issuers and, secondly, the 'blinded blacklist', i.e. the values K_i, does not depend on any value contributed by the verifier. This means that BP may precompute these values at any time, and may even be able to reuse the same values for multiple verifiers. In terms of computation and communication complexity, our proposed protocol is essentially identical to the PSI scheme in [4, Fig. 4];

It is worth mentioning that, in the context of passports inspection, the above scheme does not impose any change to already issued electronic passports compliant to the relevant standard [11]. This is because these passports already contain a signature that covers the passport number, and this signature is already read and verified by (compliant) inspection systems.

5.2 Comparison of security properties and asymptotic efficiency

Table 2 and 3 provide an overview of the asymptotic complexities of, and the privacy properties achieved by: the simple scheme discussed in Sect. 3, the 'Privacy-Preserving Revocation Checking' (PPRC) scheme proposed by Narisimha *et al.* [16], the APSI scheme of De Cristofaro and Tsudik [4, Fig. 2] with the discussed modifications in Sect. 4.2, and our proposal discussed in Sect. 5. In Table 3, $\mathcal{T}_{\mathcal{I}}$ denotes the set of token identifiers that are signed by a particular token issuer using the same signature key.

Table 4 provides a comparison in concrete computational and communicational costs specifically for the use case of checking a blacklist of passports. In this table, 'x exp' denotes that x mudular exponentiations must be computed. It is also assumed that ten million passports are blacklisted (i.e. that $|B_v| = 10^7$). This may seem to be a large number given that, according to some estimates,

Table 2. Asymptotic efficiency of existing schemes

Scheme	Init			Query		
	comp_{V}	$\text{comp}_{\mathsf{BP}}$	comm	comp_{V}	$\text{comp}_{\mathsf{BP}}$	comm
Simple scheme	-	$\mathcal{O}(\lvert B_v\rvert)$	$\mathcal{O}(\lvert B_v\rvert)$	$\mathcal{O}(\ln\ln\lvert B_v\rvert)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
PPRC	-	-	-	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\lvert T\rvert)$
APSI-1	$\mathcal{O}(1)$	$\mathcal{O}(\lvert B_v\rvert)$	$\mathcal{O}(\lvert B_v\rvert)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Our proposal	-	$\mathcal{O}(\lvert B_v\rvert)$	$\mathcal{O}(\lvert B_v\rvert)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Table 3. Privacy properties of existing schemes

Scheme	T-User Privacy	Blacklist Hiding	Online operation	Token binding
Simple scheme	Yes, for $T = B_v$	Yes	Yes	No
PPRC	Yes, with tunable T	No	Partially	No
APSI	Yes, for $T = \mathcal{T}_{\mathcal{I}}$	Yes	Yes	Weak Proactive
Our proposal	Yes, for $T = \mathcal{T}$	Yes	Yes	Weak Proactive

there are approximately five hundred million passports in circulation worldwide; we nevertheless believe that a real system must operate efficiently for blacklists of this order of magnitude. In order to provide a fair comparison between different schemes, we selected cryptographic keys sizes according to the 2011 ECRYPT II (European Network of Excellence in Cryptology) keysize report [1] for a security level that provides medium-term protection. More precisely, we assumed an RSA modulus of 2432 bits, and a bitlength of 224 for both elliptic curve elements and token identifiers.

Table 4. Comparision of existing schemes, applied to the use case of passports.

Scheme	Init			Query		
	comp_{V}	$\text{comp}_{\mathsf{BP}}$	comm	comp_{V}	$\text{comp}_{\mathsf{BP}}$	comm
Simple scheme	-	$9 \cdot 10^7$ exp	2.1 GB	6 exp	2 exp	56 B
PPRC	-	-	-	1 exp	1 exp	2.8 kB
APSI	2 exp	10^7 exp	267 MB	2 exp	1 exp	608 B
Our proposal	-	10^7 exp	267 MB	1 exp	1 exp	608 B

6 Conclusion and outlook

In this paper, we considered the problem of checking a remote blacklist in order to establish the legitimacy of a token in a privacy-preserving way, i.e. in a way that does not leak more information than strictly necessary. We compared existing schemes from the literature and described slight adaptations that tweak these schemes for optimal efficiency in the remote blacklist setting. Finally, we described how to adapt the most efficient scheme we found in the literature such that it achieves the property of 'weak proactive token binding'. This property

guarantees that the token verifier cannot query the blacklist provider about tokens that it never saw. We believe that it is possible to use our proposal with electronic passports that conform to [11]. Moreover, due to its high efficiency and moderate storage requirements, we believe that it can be integrated into all online inspection devices, both fixed and mobile.

The work in this paper is limited in several respects, as follows. Firstly, while we believe that our survey covers most recent work in the area, is far from complete. Secondly, the arguments in this paper are informal. Important future research includes the formalisation of the different security notions and adversary models, with the aim to provide security proofs. Thirdly, the construction of a scheme that simultaneously achieves strong proactive token binding, $\mathcal{T}$-user privacy and blacklist hiding, as well as the construction of efficient schemes with unconditional rather than computational privacy guarantees, is also an interesting direction of future research.

Acknowledgements

We would like to thank Julien Bringer for his insightful comments on an earlier version of this paper. This work was supported by the Flemish Government, IWT SBO SPION, FWO G.0360.11N Location Privacy, and by the Research Council KU Leuven: GOA TENSE; and by the European Commission through the FIDELITY project (contract number 284862).

References

1. Yearly Report on Algorithms and Keysizes (2011), D.SPA.17 Rev. 1.0. Technical report, ICT-2007-216676 ECRYPT II, June 2011.
2. B. Chor, N. Gilboa, and M. Naor. Private information retrieval by keywords. Cryptology ePrint Archive, Report 1998/003, 1998. `http://eprint.iacr.org/`.
3. E. D. Cristofaro, S. Jarecki, J. Kim, and G. Tsudik. Privacy-preserving policy-based information transfer. In I. Goldberg and M. J. Atallah, editors, *Privacy Enhancing Technologies, 9th International Symposium, PETS 2009, Seattle, WA, USA, August 5–7, 2009. Proceedings*, volume 5672 of *Lecture Notes in Computer Science*, pages 164–184. Springer, 2009.
4. E. D. Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In R. Sion, editor, *Financial Cryptography and Data Security, 14th International Conference, FC 2010, Tenerife, Canary Islands, January 25-28, 2010, Revised Selected Papers*, volume 6052 of *Lecture Notes in Computer Science*, pages 143–159. Springer, 2010.
5. E. D. Cristofaro and G. Tsudik. Experimenting with fast private set intersection. In S. Katzenbeisser, E. Weippl, L. J. Camp, M. Volkamer, M. K. Reiter, and X. Zhang, editors, *Trust and Trustworthy Computing - 5th International Conference, TRUST 2012, Vienna, Austria, June 13–15, 2012. Proceedings*, volume 7344 of *Lecture Notes in Computer Science*, pages 55–73. Springer, 2012.
6. E. D. Cristofaro and G. Tsudik. On the performance of certain private set intersection protocols. Cryptology ePrint Report 54, 2012.

7. S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.
8. M. J. Freedman, K. Nissim, and B. Pinkas. Efficient Private Matching and Set Intersection. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2004.
9. S. Hohenberger and S. A. Weis. Honest-verifier private disjointness testing without random oracles. In *Privacy Enhancing Technologies – PET 2006*, volume 4258 of *LNCS*, pages 277–294. Springer, 2006.
10. Y. Huang, D. Evans, and J. Katz. Private set intersection: Are garbled circuits better than custom protocols? In *NDSS 2012, Proceedings*. IEEE, 2012.
11. International Civil Aviation Organization. *Document 9303 Volume 2, Part 1*, 2006.
12. A. Juels and M. Wattenberg. A fuzzy commitment scheme. In J. Motiwalla and G. Tsudik, editors, *CCS '99, Proceedings of the 6th ACM Conference on Computer and Communications Security, Singapore, November 1–4, 1999*, pages 28–36, 1999.
13. A. Kiayias and A. Mitrofanova. Testing Disjointness of Private Datasets. In A. S. Patrick and M. Yung, editors, *Financial Cryptography and Data Security – FC 2005*, volume 3570 of *LNCS*, pages 109–124. Springer, 2005.
14. N. Li, W. Du, and D. Boneh. Oblivious signature-based envelope. *Distributed Computing*, 17(4):293–302, 2005.
15. M. Naor and B. Pinkas. Oblivious Transfer with Adaptive Queries. In M. J. Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 1999.
16. M. Narasimha, J. Solis, and G. Tsudik. Privacy-Preserving Revocation Checking. *Int. J. Inf. Secur.*, 8(1):61–75, jan 2009.
17. S. Nasserian and G. Tsudik. Revisiting oblivious signature-based envelopes. In G. D. Crescenzo and A. D. Rubin, editors, *Financial Cryptography and Data Security, 10th International Conference, FC 2006, Anguilla, British West Indies, February 27–March 2, 2006, Revised Selected Papers*, volume 4107 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2006.
18. W. Ogata and K. Kurosawa. Oblivious keyword search. *Journal of Complexity - Special issue on coding and cryptography*, 20(2-3):356–371, 2004.
19. J. Solis and G. Tsudik. Simple and Flexible Revocation Checking with Privacy. In *Proceedings of the 6th international conference on Privacy Enhancing Technologies*, PET'06, pages 351–367, Berlin, Heidelberg, 2006. Springer-Verlag.
20. Q. Ye, H. Wang, J. Pieprzyk, and X.-M. Zhang. Efficient Disjointness Tests for Private Datasets. In Y. Mu, W. Susilo, and J. Seberry, editors, *13th Australasian Conference Information Security and Privacy – ACISP 2008*, volume 5107 of *Lecture Notes in Computer Science*, pages 155–169. Springer, 2008.
21. Q. Ye, H. Wang, J. Pieprzyk, and X.-M. Zhang. Unconditionally secure disjointness tests for private datasets. *IJACT*, 1(3):225–235, 2009.