Towards the Trust-Enhancements of Single Sign-On Services

Xuhua Bao*†, Xiaokun Zhang‡, Jingqiang Lin§¶♯, Dawei Chu§¶♯, Qiongxiao Wang§¶♯, Fengjun Li◇
*Legendsec Information Technology (Beijing) Inc., CHINA
†Joint Laboratory of Institute of Information Engineering, Chinese Academy of Sciences and QiAnXin Group
‡Academy of Opto-Electronics, Chinese Academy of Sciences
§State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences
¶Data Assurance and Communication Security Research Center, Chinese Academy of Sciences
♯School of Cyber Security, University of Chinese Academy of Sciences
◇Department of Electrical Engineering and Computer Science, the University of Kansas, USA
Email: baoxuhua@qianxin.com, xkzhang@aoe.ac.cn, dwchu@cashq.ac.cn, {linjq, qxwang}@is.ac.cn, fli@ku.edu

***Abstract*—Single sign-on (SSO) becomes popular as the identity management and authentication infrastructure in the Internet. A user receives an SSO ticket after being authenticated by the identity provider (IdP), and this IdP-issued ticket enables him to sign onto the relying party (RP). However, there are vulnerabilities (e.g., Golden SAML) that allow attackers to arbitrarily issue SSO tickets and then sign onto any RP on behalf of any user. Meanwhile, several incidents of certification authorities (CAs) also indicate that the trusted third party of security services is not so trustworthy as expected, and fraudulent TLS server certificates are signed by compromised or deceived CAs to launch TLS man-in-the-middle attacks. Various approaches are then proposed to tame the absolute authority of (compromised) CAs, to detect or prevent fraudulent TLS server certificates in the TLS handshakes. The trust model of SSO services is similar to that of certificate services. So this paper investigates the defense strategies of these trust-enhancements of certificate services, and attempts to apply these strategies to SSO to derive the trust-enhancements applicable in the SSO services. Our analysis derives (*a*) some security designs which have been commonly-used in the SSO services or non-SSO authentication services, and (*b*) two schemes effectively improving the trustworthiness of SSO services, which are not widely discussed or adopted.**

***Index Terms*—Single Sign-On, Trust Management, Certificate, Public Key Infrastructure, Trusted Third Party.**

I. Introduction

Single sign-on (SSO) services have become the very popular identity management and authentication infrastructure in the Internet. For example, Google, PayPal, Facebook, Microsoft, Alibaba and Tencent provide their SSO services, which allow a user to sign onto millions of network systems with the same account using the SSO protocols such as OpenID Connect [1] and SAML with WS-Security [2] in SOAP [3]. In order to sign onto a relying party (RP), a user is first authenticated by the identity provider (IdP), and then request a ticket from the IdP (e.g., an assertion in SAML or id-token in OpenID Connect, which is signed by the IdP). This ticket is forwarded to the target RP. After verifying the validity of the SSO ticket, the RP allows the bearer of the ticket (i.e., the authenticated user) to sign on as the account enclosed in the ticket. Alternatively, in the authorization code flow of OpenID Connect [1], the authenticated user forwards a random authorization code to the target RP, and the RP uses it to obtain the signed id-token (or ticket) from the IdP.

An IdP in the SSO services takes a very similar trusted role as a certification authority (CA) in public key infrastructures (PKIs), which signs certificates to provide security guarantees (such as authentication, confidentiality, and non-repudiation) for various PKI-based applications [4]. By accepting the SSO tickets issued by an IdP, an RP deputes its user authentication to the IdP. Similarly, trusting the CAs, a browser establishes secure channels with the web servers after verifying the CA-signed TLS server certificates. At the same time, the security of IdPs in SSO services and CAs in certificate services, depends on their controls and protections of the private keys to issue SSO tickets and PKI certificates, respectively.

Since an IdP is authorized to authenticate all users for any RP trusting the IdP, it is becoming an attack target of interest [5]–[8]. Moreover, there exist vulnerabilities that allow attackers to compromise the IdP to issue SSO tickets arbitrarily. For example, in the golden SAML attack [5], the adversaries only need an unprivileged user account of Microsoft Active Directory Federation Services to access the private key to issue fraudulent SSO tickets (i.e., a verifiable ticket enclosing the account of an victim user). Then, such fraudulent SSO tickets allow the attackers to sign onto online applications on behalf of any victim user.

While it is widely recognized that no perfect operating system or software is available to provide online services, security incidents of trusted third parties are disclosed. For example, several well-known CAs, which are accredited to provide publicly-trusted certificate services, were intruded or deceived to sign fraudulent TLS server certificates [9]–[12], which contained well-formatted but incorrect or fake information. These fraudulent certificates are exploited to launch TLS man-in-the-middle (MitM) attacks to intercept and decrypt the private data of victims. These incidents of CAs imply that an online IdP might be compromised or intruded to issue fraudulent SSO tickets. Moreover, fraudulent SSO tickets are

Xiaokun Zhang is the corresponding author.

978-1-7281-2319-6/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: University of Kansas Libraries. Downloaded on August 07,2021 at 22:12:00 UTC from IEEE Xplore. Restrictions apply.

much more difficult to detect than fraudulent TLS server certificates, because they are forwarded only to a particular RP, valid for a very short period of time (e.g., 3 to 5 minutes), and transmitted over private channels such as HTTPS.

Security-enhanced mechanisms are designed to mitigate the risk by malicious or compromised third parties that were fully trusted in the past. Various schemes are proposed to tame the absolute authority of CAs in the PKI ecosystem and reduce the damages due to fraudulent TLS server certificates, including public key or certificate pinning (e.g., HPKP [13] and TACK [14]), public logging (e.g., certificate transparency [15] and ARPKI [16]), restricted scopes of services (e.g., CAge [17] and Certlock [18]), multi-path verification (e.g., DoubleCheck [19] and Perspectives [20]), subject-controlled policies (e.g., DANE [21] and PoliCert [22]), and multi-authority certification (e.g., PoliCert [22] and ARPKI [16]). These approaches are designed based on very different defense strategies, to improve the trustworthiness of certificate services.

In this paper, we attempt to follow each of the different defense strategies of these trust-enhancements for certificate services, to investigate the trust-enhancements of SSO against potentially compromised IdPs. We survey the existing trust-enhancements for certificate services, and summarize the principal defense strategies based on the detailed trust-enhancements. Then, by applying each strategy to the SSO services, we derive tentative security designs. Our analysis shows that some derived schemes have been commonly adopted to authenticate users, which may be in SSO or non-SSO scenarios. More importantly, we derive two schemes called *ticket synchronization* and *ticket transparency*, as the new candidate trust-enhancements against the potentially compromised IdPs of SSO. The advantages and shortcomings of these two schemes are also discussed.

Contribution. We summarize the defense strategies of trust-enhancements of certificate services, and apply these strategies to SSO services. Then, different schemes against fraudulent SSO tickets are derived and analyzed. Among the derived schemes, ticket synchronization is presented for the first time against fraudulent SSO tickets. To the best of our knowledge, this is the first attempt to systematically investigate the problems caused by compromised IdPs in SSO services.

II. The Trust-Enhancements of Certificate Services

Different kinds of incidents cause accredited CAs to sign fraudulent certificates, including network intrusions [9]–[12], reckless identity validations [23]–[26], mis-operations [27]–[29], or even government compulsions [30], [31]. A fraudulent TLS server certificate binds a domain name (e.g., www.facebook.com or www.hotmail.com) to a key pair held by the man-in-the-middle (MitM) attackers, but not the legitimate web server. These serious incidents in the real world imply that even an accredited signing system that is implemented in well-protected organizations, could still be compromised to sign fraudulent messages, which contain well-formatted and verifiable but incorrect or misleading data. Consequently, various trust-enhancement schemes against compromised CAs are then proposed as follows.

A. The Existing Schemes against Compromised CAs

The basic idea of pinning is that a TLS client (e.g., a browser) by itself maintains the relevant certificates or public keys of the visited domain (i.e., pins certificates or public keys locally) [32]. Pinning works at the level of HTTP or TLS. HPKP [13] enables an HTTP server to instruct the browsers to locally pin its certificates. The following certificates may be pinned for a domain [13]: (*a*) the TLS server certificates; and (*b*) the certificate(s) of the intermediate and/or root CAs to verify the server certificate. Then, the pinned certificates are used to verify the TLS server certificates in the future, in addition to the standard validation of certificate chain [33]. TACK [14] defines TLS extensions to enables a TLS server to pin another public key in clients. Then, this TACK key is used to assert the authenticity of the TLS server certificate by the server itself (i.e., sign the certificate again).

Certificate transparency [15] is proposed to improve the accountability of certificate signing. After signing a certificate, the CA submits it to a log server, and the log server responds with a signed certificate timestamp (SCT), which is a promise to record the certificate in publicly-visible logs. A TLS server presents the SCTs along with its certificate in TLS handshakes; otherwise, browsers reject the TLS server certificate. Then, the web server regularly searches for all certificates binding its domain in the public logs, to detect possible fraudulent certificates among them.

In the traditional web PKI, a CA is authorized to serve any domains. For example, a root CA certificate pre-installed in Windows [34], Apple [35] or Mozilla [36] is used to verify any TLS server certificate received in the platform. Thus, once the attackers compromise the weakest publicly-trusted CA, they could issue fraudulent certificates binding any domain. CAge [17] and Certlock [18] restrict the scopes of certificate services of a certain CA in different ways. CAge specifies the restriction rules on the set of TLDs for which each CA is assumed to issue certificates. The rules are derived based on 1.95 million valid certificates issued by more than 1,200 CAs for 2.55 million domains. Certlock enforces another restriction that the country of the CAs issuing TLS server certificates for a domain does not change in the future. The rules are enforced on TLS clients when a server certificate is being verified. Warnings are displayed to the users, when the browser receives any TLS server certificate violating the restriction rules.

Perspectives [20] and Convergence [37] introduce a set of independent notaries, which fetch and maintain the certificates (or public keys) of network services. When a client is verifying the certificate of a network service, it also retrieves the records from the notaries and compares them with the one received directly from the server. In order to eliminate the user privacy (i.e., the history of visiting activities) leaked to the notaries, DoubleCheck [19] requires the TLS client to establish extra anonymous Tor links to receive another copy of the server certificate, and compare these certificates from

Authorized licensed use limited to: University of Kansas Libraries. Downloaded on August 07,2021 at 22:12:00 UTC from IEEE Xplore. Restrictions apply.

TABLE I
THE DEFENSE STRATEGIES OF DIFFERENT TRUST-ENHANCEMENTS OF CERTIFICATE SERVICES

Defense Strategy	Description
Pinning	The certificate is maintained by the TLS clients locally.
Public logging	Each certificate is publicly-visible in the logs.
Restricted scopes of services	Each CA serves only some scopes of domains.
Multi-path verification	Multiple copies of the certificate are obtained from different network paths.
Subject-controlled policies	A domain owner specifies the list of CAs authorized to sign certificates for its domain.
Multi-authority certification	A certificate is certified and signed by multiple CAs.

different network paths. The certificate is accepted, only if the copies from multiple network paths are identical.

DANE [21] and CAA [38] allow a domain owner to specify its own certificate policy as DNS resource records. These subject-controlled certificate policies include: certification authority authorization before certificate issuance [38], and security-enhanced certificate verification based on DNS security extensions (DNSSEC) [21], [39]. The policies specify the list of CAs authorized to sign certificates for the domain, and sometimes even the detailed list of TLS server certificates issued with the domain owner's authorization.

Multi-signature certificates are proposed in PoliCert [22] and ARPKI [16]: a certificate is certified and signed by multiple independent CAs. PoliCert encodes the subject certificate policies as a multi-signature certificate, while ARPKI integrates multi-signature certificates with redundant logs, among which publicly-logged certificates are synchronized.

B. The Defense Strategies of Trust-Enhancements

Table I summarizes the principal defense strategies of different trust-enhancements. We describe these defense strategies as follows. It is worthy noting that some trust-enhancement schemes integrate several strategies, such as TACK [14], PoliCert [22] and ARPKI [16]. TACK instructs the TLS clients to pin a TACK key, to verify the extra signature of the server certificate, which can be viewed as a subject certificate policy. PoliCert and ARPKI integrate multi-signature certificates, with subject certificate policies and publicly-logged certificates, respectively.

Strategy-A: Pinning. The certificates (or public keys) of a TLS server are maintained and pinned locally by clients [13], [14]. Then, mis-matched certificates are detected or rejected by the TLS clients.

Strategy-B: Public logging. All certificates are required to be publicly-visible [15], [16], and then any fraudulent certificate will be detected by the domain owner.

Strategy-C: Restricted scopes of services. A CA is restricted to serve only some scopes of domains, and the rules are enforced in browsers [17], [18]. A certificate violating the rules will trigger browser warnings.

TABLE II
THE MAPPING BETWEEN SSO SERVICES AND CERTIFICATE SERVICES

	SSO Service	Certificate Service
Third trusted party	IdP	CA
Certified entity	Authenticated user	TLS server
Certified identity	Account	Domain
Verified message	SSO ticket	Certificate
Verifier	Target RP	Browser

Strategy-D: Multi-path verification. A server certificate received in TLS handshakes, is compared with other copies fetched from different paths such as extra links by notaries to the visited server [20], [37] or anonymous Tor links [19]. Then, the certificate is accepted only if they are identical.

Strategy-E: Subject-controlled policies. The certificate subject (or domain owner) specifies its certificate policies or confirm its certificates in different forms, such as DNS resource records [21], [38], subject-policy certificates [22], and TLS extensions [14]. A certificate violating these policies is considered as invalid or suspicious.

Strategy-F: Multi-authority certification. A certificate is confirmed and signed by multiple independent CAs [16], [22], so that a compromised CA is unable to issue fraudulent certificates arbitrarily by itself.

III. THE TRUST-ENHANCEMENTS OF SINGLE SIGN-ON

In this section, we compare the SSO services and the certificate services, and present the threat model and design goal of the trust-enhancements of SSO. Then we apply the defense strategies of the trust-enhancements of certificate services to SSO. Finally, the derived tentative trust-enhancement schemes for SSO services are analyzed one by one.

A. The Comparison of SSO Services and Certificate Services

The SSO scenario consists of an IdP, a number of users and multiple RPs. The IdP issues an SSO ticket after authenticating a user, and then the user forwards this ticket to the target RP directly or indirectly. The RP allows the bearer of an SSO ticket to sign on as the account enclosed in the ticket, after verifying the ticket. An SSO ticket typically includes: the account of the authenticated user, the identity of the target RP, the validity period, and usually a nonce against replay attacks.

In the SSO scenario, the IdP plays a similar role of trusted third party as a CA in certificate services: the verifier (i.e., the target RP in SSO services or the browser accepting TLS server certificates) depends on the information enclosed in the certified messages (i.e., SSO tickets or certificates), to finish its security functions. An RP accepts the SSO tickets as the outputs of trusted authentication services, while a browser uses the TLS server certificates to establish secure channels cryptographically. Table II lists the mapping between the components of SSO services and those of certificate services. This mapping helps us to apply the defense strategies from the certificate services to the SSO services.

The major difference is that a certificate is always valid within its validity period (e.g., one year), while an SSO

Authorized licensed use limited to: University of Kansas Libraries. Downloaded on August 07,2021 at 22:12:00 UTC from IEEE Xplore. Restrictions apply.

ticket is valid only once even within its validity period. Meanwhile, there is a little difference in the transmission of verified messages. In the certificate services, a certificate is always sent by the authenticated TLS server to browsers (i.e., from the certified entity to the verifier). On the contrary, in the SSO scenarios, an SSO ticket may be forwarded by the authenticated user to the RP, or an authorized code is forwarded to the RP which uses it to obtain the signed SSO ticket from the IdP by itself [1].

B. Design Goal and Threat Model

Security Goal. The trust-enhancements of SSO attempt to mitigate the damages by fraudulent tickets, issued by compromised IdPs. Such fraudulent tickets allow the attackers to sign onto the RPs to learn the private data or conduct some operations on behalf of the victims in the RP.

Since we cannot always prevent a compromised IdP from issuing fraudulent SSO tickets, a trust-enhancement scheme may (*a*) prevent the target RP from accepting such fraudulent tickets, and/or (*b*) detect any fraudulent ticket whether it has been accepted by the RP or not.

Threat Model. We assume that the online IdP might be compromised or deceived to issue fraudulent SSO tickets, while the RPs are always trusted because one of our design goals is to protect the private data in RPs against unauthorized users (or attackers) exploiting fraudulent tickets. A malicious RP could access these private data arbitrarily, so we have to assume trusted RPs.

The authenticated legitimate user acts as specified in the SSO protocols to forward SSO tickets to the target RP. We do not specially consider the attacks on legitimate users, such as password guessing and IdP cookie hijacking [6]. The RP cannot distinguish such attacks from the one caused by compromised IdPs, because these attacks indistinguishably lead to verifiable but fraudulent SSO tickets. A verifiable SSO ticket, either requested by an attacker on behalf of the compromised user or issued by a compromised IdP, enables the bearer to sign onto the target RP. Section IV specially compares the attacks by compromised IdPs with the ones due to stolen user authenticators.

C. Applying Different Defense Strategies in SSO Services

We apply the defense strategies of trust-enhancements for certificate services in Table I to the SSO scenarios, following the mapping in Table II, to derive the trust-enhancements. In fact, some derived solutions have been proposed and deployed in SSO scenarios or even non-SSO authentication services, some of which are designed against the attacks directly compromising the legitimate user.

Strategy-A: Pinning. The certified entity and the verifier (i.e., the RP and the user) maintain the tickets by themselves. Thus, the RP synchronizes a secret sequence number with each user. The sequence number is initialized by the RP and the user, without the participation of IdPs. The user locally holds this secret number, and it is sent along with the IdP-issued SSO ticket to the target RP. The RP verifies the ticket and additionally compares the numbers. The sequence number is updated after each successful sign-on, since each SSO ticket is valid only once at most while the pinning of a certificate is updated after it expires.

In the remainder, we call this scheme *ticket synchronization*, and it is explained with more details in Section III-D.

Strategy-B: Public logging. Similar to publicly-logged certificates, each ticket accepted by the RP is logged in public logs, especially visible to the authenticated user, the account of which is enclosed in the ticket. Redundant and fault-tolerant log servers are needed to record each ticket, similarly to the log servers in certificate transparency [15]. However, it has to carefully protect the user privacy information in SSO tickets, because each ticket contains privacy information such as the service requester (i.e., the user), the service provider (i.e., the RP), the occurrence time of sign-on activities, etc.

We call it *ticket transparency*, and Section III-E explains this scheme with more details.

Strategy-C: Restricted scopes of services. It means that the service scope of an IdP is restricted. That is, an IdP is allowed to issue SSO tickets only for certain groups of users, and/or accepted by certain RPs.

In fact, this restriction has currently been enforced commonly in SSO services. For an RP, there is no trusted IdP by default. Different from a certificate signed by publicly-trusted CAs [34]–[36] and then accepted by most mainstream browsers, an SSO ticket is accepted by an RP only after the RP explicitly configures the list of trusted IdPs [1], [8].

Strategy-D: Multi-path verification. An SSO ticket is compared with another copy obtained from a different network path, before it is accepted by the RP. On receiving an SSO ticket from somebody, the RP attempts to obtain another copy of the ticket by actively connecting to the user, the account of which is enclosed in the ticket, from a different network path. Then, two copies of the ticket will be compared.

This scheme is not so practical in the SSO scenarios to verify each SSO ticket, because it is usually time-consuming for the RP to establish another extra connection to the user. Sometimes it is even impossible to establish such connections to a user, because the user is usually not bound to any global IP address.

Similar schemes have been proposed and adopted against stolen user authenticators as follows. For example, when an authentication service suspects that some user account has been compromised and an attacker is signing on, the online service will send a private hyperlink to the email address or a random code to the mobile-phone number, both of which are pre-configured by the user. Then, after the legitimate user confirms this sign-on by clicking the hyperlink or submitting the random code, the sign-on succeeds.

Strategy-E: Subject-controlled policies. Each user defines its ticket policy (i.e., the list of IdPs authorized to issue tickets for it), and the policy is configured in the RPs. Any ticket violating the policy triggers alerts in the RP.

Such policies are inherently effective in the SSO services,

Authorized licensed use limited to: University of Kansas Libraries. Downloaded on August 07,2021 at 22:12:00 UTC from IEEE Xplore. Restrictions apply.

and an IdP is authorized to issue SSO tickets only for its user account. For example, the Google SSO service never issues tickets for a Facebook account. In fact, similar strategies are usually enforced by the RPs (but not defined by the users), but with more intelligent policies. For example, the RP automatically learns the pattern of each user's sign-on activities (e.g., when, from where and how often it signs onto the RP), and requires extra authentication actions or even rejects the sign-on once any sign-on attempt violates the pattern. Besides, the most typical extra authentication action is the verification via the extra path, as described above. This strategy is also commonly adopted in non-SSO network applications, against stolen user authenticators.

Strategy-F: Multi-authority certification. It means that the RP requires that a user shall be authenticated by multiple IdPs. That is, an SSO ticket is issued by multiple independently-operated IdPs cooperatively.

It is not user-friendly, for a user has to be authenticated by multiple IdPs. In practice, this strategy is sometimes implemented as another form – multi-factor authentication (MFA). It is worthy noting that MFA does not always work effectively against a compromised IdP, because the compromised IdP may issue tickets even when the "user" is authenticated by no factor, unless some authentication factor is verified by the RP or another independent IdP.

Potential user-friendly implementations of multi-authority certification in the SSO scenarios against the compromised IdP, are as follows. A user authenticates itself to the first IdP (e.g., by the user password), to sign onto the target RP. When there are critical operations, the RP (*a*) requires the user to present another authentication factor which is verified by the RP, or (*b*) redirects the user to another IdP.

Summary. Among these 6 schemes derived from the very different defense strategies against fraudulent certificates, we find that 4 schemes have been commonly adopted, while some of them are not designed specially for the SSO scenarios and some are proposed against compromised user accounts. Ticket synchronization derived from pinning and ticket transparency derived from public logging, have not been commonly adopted or comprehensively investigated. So, in the next sections, we analyze these two schemes with more details.

D. Ticket Synchronization

A user synchronizes a secret sequence number with each RP, typically as a long-term cookie. It is kept by the user and the RP, but not by the IdP. This cookie is submitted to the RP along with the SSO ticket, when the user is signing onto the RP. Then, after verifying the SSO ticket, the RP will compare the submitted sequence number with the one it stores, and the user is allowed to sign on only if they match. Moreover, the sequence number is updated (e.g., increased by a random value), after each successful sign-on.

Ticket synchronization follows the assumption of trust on first use, as other trust-enhancement designs (especially pinning-based solutions) of certificate services [13], [14], [20]. This sequence number is initialized randomly, when a user registers in the RP and signs on for the first time. Then, it is synchronized between the user and the RP as above. Due to the great amount of users, the RP will keep these sequence numbers in non-volatile storage, along with other user configurations.

Ticket synchronization is straightforward and effective against fraudulent tickets. An attacker attempting to sign on with a fraudulent ticket, will be detected by the RP because the attacker does not hold the correct sequence numbers. However, a legitimate user has to use the same device that keeps the synchronized number, to sign on the RPs; otherwise, it will be falsely detected as an attacker.

False positives of ticket synchronization happen when a legitimate user is signing on from a different device. So the RP may initiate extra authentication actions (e.g., send a random code to the pre-configured mobile-phone number and wait for the user to input this code), when the sequence numbers mismatch. If this extra stronger authentication fails, the sign-on attempt is rejected. If the stronger authentication succeeds, the authenticated user may choose to sign on the RP always from this new device in the future, and then a new sequence number is initialized between the user device and the RP.

E. Ticket Transparency

Ticket transparency is proposed recently to record all SSO tickets in public logs [40], and a ticket is accepted by the RPs only if it is recorded in public logs. An extra signature by the log server is sent along with the ticket, and this extra signature is a guarantee to make the ticket be publicly visible in the logs. Then, a user is enabled to search for all tickets with his account in the public logs and detect fraudulent ones among them in the future. Meanwhile, some mechanisms are designed to ensure the correct behaviors of log servers [40]; that is, the logs are append-only, and each ticket accepted by the RPs corresponds to some entry in the logs.

Compared with certificate transparency [15], ticket transparency must carefully deal with the user privacy information in SSO tickets. The log server can adopt a blind signature scheme to additionally sign the SSO tickets [40], so that the ticket content is not disclosed to the log server. Meanwhile, a Bloom filter is used to generate the non-unique pseudonym for a user, stored along with each blindly-signed ticket in public logs. Therefore, when a user suspects there exists some fraudulent ticket labeled with his account in the public logs, it attempts to un-blind the ticket entries stored with his pseudonym, sometimes with the trusted coordinator's help. The trusted coordinator is able to un-blind all tickets in the public logs, because the blinding factors are encrypted using identity-based encryption (IBE) and the master key of IBE is held by the coordinator.

Although blind signatures are integrated in the proposed scheme of ticket transparency [40] to protect the user privacy information, there are still some user privacy leaked through the public logs. The attackers may learn a user's history of sign-on activities (only the occurrence but no information

Authorized licensed use limited to: University of Kansas Libraries. Downloaded on August 07,2021 at 22:12:00 UTC from IEEE Xplore. Restrictions apply.

about the RP signed onto) but with a false positive rate, because the pseudonym is non-unique. Thus, in the future more privacy-preserving techniques [41]–[43] shall be integrated in ticket transparency to protect user privacy better.

IV. Discussion

All trust-enhancements against fraudulent tickets issued by a compromised IdP, are also effective against the attacks exploiting stolen user authenticators (e.g., password guessing). Such trust-enhancement schemes try to guarantee that any SSO ticket is issued with the legitimate user's participation, and then an SSO ticket which is issued (*a*) by a compromised IdP or (*b*) to an attacker with stolen user authenticators, will be rejected or detected. So all schemes discussed in Section III-C, including ticket synchronization and ticket transparency, may be applied to mitigate the risk of compromised user accounts.

On the other hand, the defenses for compromised user accounts are not always effective to detect or prevent fraudulent tickets by compromised IdPs. For example, authentication revocation [6] is proposed to mitigate the damages caused by the IdP cookie hijacking in the SSO scenarios. But it depends on the request of single sign-off sent from the IdP to RPs, so it does not work if the IdP was malicious. That is, the countermeasures against potentially-compromised IdPs are implemented on trusted RPs (but not the IdP), while the designs mitigating the risk of compromised user accounts may still assume a trusted IdP (and also trusted RPs).

V. Related Work

A. Security of SSO Services

Software vulnerabilities are discovered in the implementations of different SSO protocols, including SAML with WS-Security [44], OAuth [45]–[47], OpenID and customized SSO protocols [48], some of which were deployed as widely-used SSO services in the Internet, such as Facebook OAuth [48], [49], Windows LiveID [50] and Google OpenID [48], [51]. An attacker could exploit these vulnerabilities to sign onto RPs on behalf of other users or access private information of others. Mainka et al. introduce a malicious IdP to vulnerable RPs in the discovery phase of OpenID, and then all accounts on the deceived RP are compromised [8]. [6] utilizes the design of single sign-off in OpenID Connect to revoke the access to the RPs, when the user's IdP cookie is hijacked. The above works comprehensively investigate the secure implementation and/or deployment of SSO services, and these solutions cannot deal with the problems caused by potentially compromised IdPs.

Different kinds of anonymity are introduced in the SSO scenarios. Some schemes allow a user to access RPs without revealing its identity to the RPs, based on broadcast encryption [52], group signatures [53] or extended Chebyshev Chaotic Maps [54]. [55] improves these anonymous SSO schemes, where only the target RP is able to verify the tickets, and no identity is released to the IdP. On the other hand, SPRESSO [56] and BrowserID [57] protect the user privacy from a different way – it hides the target RP from the IdP while allowing the user to request an SSO ticket to this RP with a fixed identity. The user privacy leakage due to SSO (or identity federation) is discussed [58]. Curious-but-honest IdPs are assumed in these approaches, while we try to handle the problem of fraudulent tickets issued by a compromised IdP.

B. Security of Third-Party Services

The private key generator (PKG) of IBE is responsible for generating the private keys for all users, and the private key generation is considered as another typical security service. The inherent trust on the PKG is mitigated as below: (*a*) distributed PKGs [59], [60] – the absolute authority of PKG is distributed among several independent components, and each private key generation is conducted by these components cooperatively, and (*b*) accountable IBE [61], [62] – if the PKG re-generates the private key for anybody, a proof will be produced automatically. Distributed PKGs follow the strategy of multi-authority certification, and accountable IBE works similarly to public logging. Besides, KGC-anonymous IBE [63] reduces the trust on PKGs in a different way: the PKG generates the private keys for a list of users without knowing the identity of each user.

Verifiable searchable encryption is proposed to verify the correctness and completeness of search results from the semi-untrusted third party over encrypted data [64], [65] – some are designed for searchable symmetric encryption [66], [67], and the others are proposed for searchable public-key encryption [68], [69]. Verifiable computation on outsourced data enables a remote user to outsource the execution of a program, while providing a verifiable guarantee of integrity [70]–[73].

Independent auditors are introduced to check the integrity of data outsourced in untrusted cloud systems, while the data are not disclosed to the auditors [74]–[76]. Liu et al. presented consistency as a service [77]: a data cloud is maintained by the cloud service provider, and a group of users constitute an audit to verify whether the data cloud provides the promised level of consistency. PDP [78] and POR [79] enable the remote tenants to verify whether the data are intact in the untrusted clouds, with lightweight complexity of communications and computations. [80] detects CPU cheating on virtual machines maintained by semi-trusted cloud providers, using CPU-intensive calculations.

Third-party monitors [81]–[86] implement the monitoring functions of certificate transparency for users: a third-party monitor fetches all certificates from public logs, and provides certificate search services. Then, a domain owner searches all certificates issued for its domain from the third-party services. The recent study [87] shows that these third-party monitors do not perform reliably as expected and the search results may not return complete certificates. [88] finds that the TLS/HTTPS configurations of these third-party monitors are not strong enough, and then the vulnerabilities might be exploited to manipulate the certificate search results.

The SSO services can be viewed as another kind of third-party security services – ID as a service, but the problem of compromised IdPs have not been well investigated in the literature. Our work shares the same spirit with the above

Authorized licensed use limited to: University of Kansas Libraries. Downloaded on August 07,2021 at 22:12:00 UTC from IEEE Xplore. Restrictions apply.

studies that the trustworthiness assumption of a third party needs to be reduced.

VI. Conclusion and Future Work

The SSO systems provide identity management and authentication services in the Internet. An SSO ticket issued by an IdP allows a user to sign onto many RPs trusting the IdP, on behalf of the account labeled in the ticket. As a typical third-party security service provider, the IdP is becoming an attractive target of interests to attackers. Meanwhile, there are vulnerabilities allowing the attackers to arbitrarily issue SSO tickets. Therefore, we need to design mechanisms to detect and/or prevent fraudulent SSO tickets issued by potentially compromised IdPs.

In this paper, we apply the principal defense strategies of the trust-enhancements of certificate services, which are originally proposed against fraudulent TLS server certificates, to the SSO scenarios. Then, different schemes mitigating the risk of fraudulent SSO tickets, are derived from the defense strategies. Some schemes have been proposed and deployed against compromised accounts, while the others are discussed for the first time. In the future, we will conduct more comprehensive evaluations on these schemes.

References

[1] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and M. Chuck, "OpenID Connect Core 1.0," 2014, http://openid.net/specs/openid-connect-core-1_0.html.

[2] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, "OASIS standard - Web services security: SOAP message security 1.1," 2006.

[3] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. Nielsen, A. Karmarkar, and Y. Lafon, "W3C recommendation - SOAP version 1.2 part 1: Messaging framework (2nd edition)," 2007.

[4] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "IETF RFC 5280: Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," 2008.

[5] S. Reiner, "Golden SAML: Newly discovered attack technique forges authentication to cloud apps," 2017, https://www.cyberark.com/threat-research-blog/golden-saml-newly-discovered-attack-technique-forges-authentication-cloud-apps/.

[6] M. Ghasemisharif, A. Ramesh, S. Checkoway, C. Kanich, and J. Polakis, "O single sign-off, where art thou? An empirical analysis of single sign-on account hijacking and session management on the Web," in *27th USENIX Security Symposium*, 2018.

[7] S.-T. Sun, E. Pospisil, I. Muslukhov, N. Dindar, K. Hawkey, and K. Beznosov, "What makes users refuse web single sign-on? An empirical investigation of OpenID," in *7th Symposium on Usable Privacy and Security (SOUPS)*, 2011.

[8] C. Mainka, V. Mladenov, and J. Schwenk, "Do not trust me: Using malicious IdPs for analyzing and attacking single sign-on," in *1st IEEE European Symposium on Security and Privacy (Euro S&P)*, 2016.

[9] Comodo Group Inc., "Comodo report of incident," 2011, https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html.

[10] GlobalSign. (2011) Security incident report. Https://www.globalsign.com/resources/globalsign-security-incident-report.pdf.

[11] VASCO Data Security International Inc., "DigiNotar reports security incident," 2011, https://www.vasco.com/about-vasco/press/2011/news_diginotar_reports_security_incident.html.

[12] B. Morton, "More Google fraudulent certificates," 2014, https://www.entrust.com/google-fraudulent-certificates/.

[13] C. Evans, C. Palmer, and R. Sleevi, "IETF RFC 7469 - Public key pinning extension for HTTP," 2015.

[14] M. Marlinspike, "Trust assertions for certificate keys," 2013, http://tack.io/draft.html.

[15] B. Laurie, A. Langley, and E. Kasper, "IETF RFC 6962 - Certificate transparency," 2014.

[16] D. Basin, C. Cremers, T. Kim, A. Perrig, R. Sasse, and P. Szalachowski, "ARPKI: Attack resilient public-key infrastructure," in *21th ACM Conference on Computer and Communications Security (CCS)*, 2014.

[17] J. Kasten, E. Wustrow, and A. Halderman, "CAge: Taming certificate authorities by inferring restricted scopes," in *17th Financial Cryptography and Data Security Conference (FC)*, 2013.

[18] C. Soghoian and S. Stamm, "Certified lies: Detecting and defeating government interception attacks against SSL," in *15th Financial Cryptography and Data Security Conference (FC)*, 2012.

[19] M. Alicherry and A. Keromytis, "Doublecheck: Multi-path verification against man-in-the-middle attacks," in *14th IEEE Symposium on Computers and Communications (ISCC)*, 2009.

[20] D. Wendlandt, D. Andersen, and A. Perrig, "Perspectives: Improving SSH-style host authentication with multi-path probing," in *USENIX Annual Technical Conference (ATC)*, 2008, pp. 321–334.

[21] P. Hoffman and J. Schlyter, "IETF RFC 6698 - The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA," 2012.

[22] P. Szalachowski, S. Matsumoto, and A. Perrig, "PoliCert: Secure and flexible TLS certificate management," in *21st ACM Conference on Computer and Communications Security (CCS)*, 2014, pp. 406–417.

[23] Microsoft, "MS01-017: Erroneous VeriSign-issued digital certificates pose spoofing hazard," 2001, https://technet.microsoft.com/library/security/ms01-017.

[24] SSL Shopper, "SSL certificate for mozilla.com issued without validation," 2008, https://www.sslshopper.com/article-ssl-certificate-for-mozilla.com-issued-without-validation.html.

[25] K. Wilson, "Distrusting new CNNIC certificates," 2015, https://blog.mozilla.org/security/2015/04/02/distrusting-new-cnnic-certificates/.

[26] Start Commercial (StartCom) Limited, "Critical event report," 2008, https://blog.startcom.org/wp-content/uploads/2009/01/ciritical-event-report-12-20-2008.pdf.

[27] B. Morton, "Public announcements concerning the security advisory," 2013, https://www.entrust.com/turktrust-unauthorized-ca-certificates.

[28] M. Zusman, "Criminal charges are not pursued: Hacking PKI," 2009, https://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-zusman-hacking_pki.pdf.

[29] A. Langley, "Further improving digital certificate security," 2013, https://security.googleblog.com/2013/12/further-improving-digital-certificate.html.

[30] P. Eckersley, "A Syrian man-in-the-middle attack against Facebook," 2011, https://www.eff.org/deeplinks/2011/05/syrian-man-middle-against-facebook.

[31] C. Soghoian and S. Stamm, "Certified lies: Detecting and defeating government interception attacks against SSL," in *15th Financial Cryptography and Data Security Conference (FC)*, 2012, pp. 250–259.

[32] A. Langley, "Public key pinning," 2011, https://www.imperialviolet.org/2011/05/04/pinning.html.

[33] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "IETF RFC 5280: Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," 2008.

[34] Microsoft Corporation, "Microsoft trusted root certificate program: Participants," 2018, https://gallery.technet.microsoft.com/Trusted-Root-Program-d17011b8.

[35] Apple Inc, "Lists of available trusted root certificates in macOS," 2018, https://support.apple.com/en-us/HT202858.

[36] Mozilla, "Mozilla included CA certificate list," 2018, https://wiki.mozilla.org/CA/Included_Certificates.

[37] M. Marlinspike, "Convergence," 2011, https://github.com/moxie0/Convergence.

[38] P. Hallam-Baker and R. Stradling, "IETF RFC 6844 - DNS certification authority authorization (CAA) resource record," 2013.

[39] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "IETF RFC 4033 - DNS security introduction and requirements," 2005.

[40] D. Chu, J. Lin, F. Li, X. Zhang, Q. Wang, and G. Liu, "Ticket transparency: Accountable single sign-on with privacy-preserving public logs," in *15th International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2019.

[41] S. Eskandarian, E. Messeri, J. Bonneau, and D. Boneh, "Certificate transparency with privacy," in *17th International Symposium on Privacy Enhancing Technologies (PETS)*, 2017.

Authorized licensed use limited to: University of Kansas Libraries. Downloaded on August 07,2021 at 22:12:00 UTC from IEEE Xplore. Restrictions apply.

[42] M. Melara, A. Blankstein, J. Bonneau, E. Felten, and M. Freedman, "CONIKS: Bringing key transparency to end users," in *24th USENIX Security Symposium*, 2015.

[43] R. Peeters and T. Pulls, "Insynd: Improved privacy-preserving transparency logging," in *21st European Symposium on Research in Computer Security (ESORICS)*, 2016.

[44] J. Somorovsky, A. Mayer, J. Schwenk, M. Kampmann, and M. Jensen, "On breaking SAML: Be whoever you want to be," in *21st USENIX Security Symposium*, 2012.

[45] S.-T. Sun and K. Beznosov, "The devil is in the (implementation) details: An empirical analysis of OAuth SSO systems," in *19th ACM Conference on Computer and Communications Security (CCS)*, 2012.

[46] H. Wang, Y. Zhang, J. Li, and D. Gu, "The Achilles heel of OAuth: A multi-platform study of OAuth-based authentication," in *32nd Annual Computer Security Applications Conference (ACSAC)*, 2016.

[47] H. Wang, Y. Zhang, J. Li, H. Liu, W. Yang, B. Li, and D. Gu, "Vulnerability assessment of OAuth implementations in Android applications," in *31st Annual Computer Security Applications Conference (ACSAC)*, 2015.

[48] R. Wang, S. Chen, and X. Wang, "Signing me onto your accounts through Facebook and Google: A traffic-guided security study of commercially deployed single-sign-on web services," in *33rd IEEE Symposium on Security and Privacy (S&P)*, 2012.

[49] Y. Zhou and D. Evans, "SSOScan: Automated testing of web applications for single sign-on vulnerabilities," in *23rd USENIX Security Symposium*, 2014.

[50] R. Wang, Y. Zhou, S. Chen, S. Qadeer, D. Evans, and Y. Gurevich, "Explicating SDKs: Uncovering assumptions underlying secure authentication and authorization," in *22nd USENIX Security Symposium*, 2013.

[51] W. Li and C. Mitchell, "Analysing the security of Google's implementation of OpenID Connect," in *13th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2016.

[52] J. Han, Y. Mu, W. Susilo, and J. Yan, "Anonymous single-sign-on for n designated services with traceability," in *6th International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2010.

[53] J. Wang, G. Wang, and W. Susilo, "Anonymous single sign-on schemes transformed from group signatures," in *5th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, 2013.

[54] T.-F. Lee, "Provably secure anonymous single-sign-on authentication mechanisms using extended Chebyshev Chaotic Maps for distributed computer networks," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1499–1505, 2018.

[55] J. Han, L. Chen, S. Schneider, H. Treharne, and S. Wesemeyer, "Anonymous single-sign-on for n designated services with traceability," in *23rd European Symposium on Research in Computer Security (ESORICS)*, 2018.

[56] D. Fett, R. Kusters, and G. Schmitz, "SPRESSO: A secure, privacy-respecting single sign-on system for the Web," in *22nd ACM Conference on Computer and Communications Security (CCS)*, 2015.

[57] Mozilla, "Application services - Firefox accounts," 2019, https://mozilla.github.io/application-services/docs/accounts/welcome.html.

[58] P. Grassi, J. Richer, S. Squire, J. Fenton, E. Nadeau, N. Lefkovitz, J. Danker, Y.-Y. Choong, K. Greene, and M. Theofanos, "SP 800-63C - Digital identity guidelines - Federation and assertions," National Institute of Standards and Technology, Tech. Rep., 2017.

[59] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology - Crypto*, 2001.

[60] A. Kate and I. Goldberg, "Distributed private-key generators for identity-based cryptography," in *7th International Conference on Security and Cryptography for Networks (SCN)*, 2010.

[61] V. Goyal, "Reducing trust in the PKG in identity-based cryptosystems," in *Advances in Cryptology - Crypto*, 2007.

[62] V. Goyal, S. Lu, A. Sahai, and B. Waters, "Black-box accountable authority identity-based encryption," in *15th ACM Conference on Computer and Communications Security (CCS)*, 2008.

[63] S. Chow, "Removing escrow from identity-based encryption," in *12th International Conference on Practice and Theory in Public Key Cryptography (PKC)*, 2009.

[64] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *21st IEEE Symposium on Security and Privacy (S&P)*, 2000.

[65] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[66] A. Soleimanian and S. Khazaei, "Publicly verifiable searchable symmetric encryption based on efficient cryptographic components," *Designs, Codes and Cryptography*, vol. 87, no. 1, pp. 123–147, 2019.

[67] R. Cheng, J. Yan, C. Guan, F. Zhang, and K. Ren, "Verifiable searchable symmetric encryption from indistinguishability obfuscation," in *10th ACM Symposium on Information, Computer and Communications Security (AsiaCCS)*, 2015.

[68] Z. Wan and R. Deng, "VPSearch: Achieving verifiability for privacy-preserving multi-keyword search over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 15, no. 6, pp. 1083–1095, 2018.

[69] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, T. Hou, and H. Li, "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 3025–3035, 2014.

[70] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *34th IEEE Symposium on Security and Privacy (S&P)*, 2013.

[71] C. Costello, C. Fournet, J. Howell, M. Kohlweiss, B. Kreuter, M. Naehrig, B. Parno, and S. Zahur, "Geppetto: Versatile verifiable computation," in *36th IEEE Symposium on Security and Privacy (S&P)*, 2015.

[72] D. Fiore, C. Fournet, E. Ghosh, M. Kohlweiss, O. Ohrimenko, and B. Parno, "Hash first, argue later: Adaptive verifiable computations on outsourced data," in *23rd ACM Conference on Computer and Communications Security (CCS)*, 2016.

[73] R. Gennaro, "Verifiable outsourced computation: A survey," in *36th ACM Symposium on Principles of Distributed Computing (PODC)*, 2017.

[74] C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[75] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *IEEE INFOCOM*, 2010.

[76] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[77] Q. Liu, G. Wang, and J. Wu, "Consistency as a service: Auditing cloud consistency," *IEEE Transactions on Network and Service Management*, vol. 11, no. 1, pp. 25–35, 2014.

[78] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Perterson, and D. Song, "Provable data possession at untrusted stores," in *14th ACM Conference on Computer and Communication Security (CCS)*, 2007.

[79] K. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in *ACM Workshop on Cloud Computing Security (CCSW)*, 2009.

[80] R. Houlihan, X. Du, C.-C. Tan, J. Wu, and M. Guizani, "Auditing cloud service level agreement on VM CPU speed," in *IEEE International Conference on Communications (ICC)*, 2014.

[81] University of Michigan, "Censys," 2018, https://censys.io/.

[82] Comodo CA Limited, "crt.sh: Certificate search," 2018, https://crt.sh.

[83] Entrust Datacard Corporation, "Certificate transparency search tool," 2018, https://www.entrust.com/ct-search/.

[84] Facebook Inc, "Facebook: Certificate transparency monitoring," 2018, https://developers.facebook.com/tools/ct/search/.

[85] Google Inc, "Google: HTTPS encryption on the web," 2018, https://transparencyreport.google.com/https/certificates.

[86] Opsmate Inc, "SSLMate: Cert spotter," 2018, https://sslmate.com/certspotter/.

[87] B. Li, J. Lin, F. Li, Q. Wang, Q. Li, J. Jing, and C. Wang, "Certificate transparency in the wild: Exploring the reliability of monitors," in *26th ACM Conference on Computer and Communications Security (CCS)*, 2019.

[88] B. Li, D. Chu, J. Lin, Q. Cai, C. Wang, and L. Meng, "The weakest link of certificate transparency: Exploring the TLS/HTTPS configurations of third-party monitors," in *18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2019.

Authorized licensed use limited to: University of Kansas Libraries. Downloaded on August 07,2021 at 22:12:00 UTC from IEEE Xplore. Restrictions apply.