Trustworthy Information: Concepts and Mechanisms

Shouhuai Xu[1], Haifeng Qian[1], Fengying Wang[1], Zhenxin Zhan[1], Elisa Bertino[2], Ravi Sandhu[1]

University of Texas at San Antonio (shxu@cs.utsa.edu)

Purdue University (bertino@cs.purdue.edu)

The Problem

Suppose Alice receives a piece of information (e.g., a message from someone or a response from a database she queried).

- To what extent should she trust the piece of information?
- Can she treat the piece of information as trustworthy? when the information is digitally signed, or when the database is maintained by a recognized organization.

The answer is NO due to the following reasons:

1. cryptographic credentials (e.g., private signing keys) can be compromised without being revoked, even possibly after a long period of time;
2. the piece of information itself was obtained from another party without proper trustworthiness guarantees;
3. the database was manipulated by an attacker.

“Trustworthy information” or “information trustworthiness management”

State of the Art. The need for “trustworthy information” or “information trustworthiness management”, is a missing piece of traditional approaches to data and information sharing.

What We Need? Information trustworthiness management should empower information consumers to justify or evaluate the trustworthiness of information, ideally in a real-time fashion.

Our Paper. This work is a significant first step towards addressing the problem.

Our Contribution

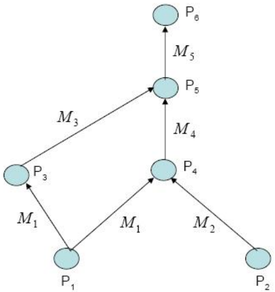
Concepts. We propose the concept of "information trustworthiness management" in the context of information networks.

Our Approach. We formulate the abstraction of "trustworthiness graph" with respect to a piece of information.

Two Mechanisms. Two mechanisms are proposed that is needed for managing trustworthiness graphs.

1. We identify a new kind of cryptographic primitive we call "provenance digital signatures" which preserves the history of a message and give an efficient construction for it.
2. We identify the need of optimal security hardening and show that the algorithmic problem in question is NP-hard, but has a good approximation algorithm.

A Simple Example



$P_1, \ldots, P_6$ in the graph represent principals, and the arcs indicate how information has moved in the information network.

Suppose P_1 enters message M_1 at time T_1 and P_2 enters message M_2 at time T_2. At time T_3, P_3 receives M_1 from P_1 and processes M_1 to produce M_3. At time T_4, P_4 receives M_1 and M_2 from P_1 and P_2, respectively, and then produce M_4. At time T_5, P_5 receives M_3 and M_4 from P_3 and P_4, respectively, and processes them to produce M_5. Finally, P_6 receives M_5 at time T_6.

Definitions

Definition (information network)

Let $[T_1, T_2]$ be a time interval and $\mathrm{V}([T_1, T_2])$ be a set of principals (users, organizations) which exchanged information during $[T_1, T_2]$. An information network over $[T_1, T_2]$ and $\mathrm{V}([T_1, T_2])$, denoted as $\mathrm{G}([T_1, T_2])$, is a pair $(\mathrm{V}([T_1, T_2]), \mathrm{E}([T_1, T_2]))$, where $\mathrm{E}([T_1, T_2])$ is the set of edges. An edge $(u, v) \in \mathrm{E}([T_1, T_2])$ if $u \in \mathrm{V}([T_1, T_2])$ has sent a message to $v \in \mathrm{V}([T_1, T_2])$ during $[T_1, T_2]$.

Definition (trustworthiness graph of an information network)

Let $[T_1, T_2]$ be time interval and T be a time instant, where $T_1 \leq T \leq T_2$. A trustworthiness graph $G(T) = (V(T), E(T))$ at time T is defined as $\mathrm{G}(\mathrm{V}[T_1, T]) = (\mathrm{V}([T_1, T]), \mathrm{E}([T_1, T]))$ with the following annotations. If $(u, v) \in E(T)$ we say that u is an "upstream" node of v and v is a "downstream" node of u. Moreover, each $(u, v) \in E(T)$ is annotated with a pair $(w_T(u, v), \theta_T(u, v))$, where $w_T(u, v) \in [0, 1]$ is v's trustworthiness evaluation of u at time T (e.g., based on the trustworthiness of information it has so-far received from u), and $\theta_T(u, v) \in [0, 1]$ is a threshold specified by v.

Definition (most/least trustworthy path)

Given a trustworthiness graph $G(T) = (V(T), E(T))$ with annotations and a path $p = (v_1, \ldots, v_\ell)$, we can define the trustworthiness of path p as $W_T(p) = \prod_{i=1}^{\ell-1} w_T(v_i, v_{i+1})$, [a] which is a real number in the interval $[0, 1]$. For a given pair of nodes $(u, v) \in V(T) \times V(T)$, let $P_T = \{(u, \ldots, v)\}$ denote the set of paths from u and v. We say that path $\bar{p} \in P_T$ is (one of) the most trustworthy if $W_T(\bar{p}) = \max\{W_T(p) : p \in P_T\}$ and path $\underline{p} \in P$ is (one of) the least trustworthy if $W_T(\underline{p}) = \min\{W_T(p) : p \in P_T\}$.

[a] This specific mathematical function is used just as an example. More sophisticated definitions are possible.

Provence Signatures

Definition (provenance signature)

A provenance signature scheme for N signers $\mathcal{S} = \{P_i : i = 1, \cdots, N\}$ (where N is polynomial in the security parameter k) consists of the following algorithms:

- Setup(1^k): is a randomized algorithm that takes as input a security parameter k and produces a set of system-wide public parameters pp.
- Keygen(pp): is a probabilistic algorithm that, on input of public parameters pp, outputs a signer's private-public key-pair (sk, pk).
- GraphCom(pp, loc, $\{G_\lambda\}_{\lambda \in \mathcal{R}}$): is an algorithm that, on input of public parameters pp, a local information string loc and $\{G_\lambda\}_{\lambda \in \mathcal{R}}$ where $\mathcal{R}$ is the group of signers who send their messages/signatures to the present signer, outputs a graph G.

Definition (To be continue)

- PSign(pp, sk_i, loc, $\{\Sigma_\lambda\}$): on input of public parameters pp, a local information string loc, a private key sk_i of P_i and each $\Sigma_\lambda = (G_\lambda, \sigma_\lambda)$ from $P_\lambda \in \mathcal{R}_i \subset \mathcal{S}$ where σ_λ is a provenance signature for G_λ generated by signer P_λ, this (possibly probabilistic) algorithm outputs a provenance signature $\Sigma = (G, \sigma)$ where $G \leftarrow$ GraphCom(pp, loc, $\{G_\lambda\}_{\lambda \in \mathcal{R}_i}$), or $\perp$ if the input $\{\Sigma_\lambda\}$ is deemed invalid.
- PVrf(pp, Σ): given parameters pp, $\Sigma = (G, \sigma)$ where G encodes a network topology graph which contains the signers' identities (or public keys) and other information, this deterministic algorithm outputs 0 if Σ is invalid; otherwise 1.

We require the scheme to have the following *correctness* property. For any sufficiently large security parameters k and system-wide parameters pp output by Setup(1^k), for all pairs of private/public key pairs $\{(sk_i, pk_i)\}_{i \in [1,N]}$ produced by Keygen(pp), and for any network topology graph G, we require $\Pr[\text{PVrf}(pp, \Sigma) = 1] = 1$ for any Σ generated by the signing algorithm. We also require that $\perp \leftarrow$ PSgin($sk_i, m, \{\Sigma_\lambda\}$) if PVrf($pp, \Sigma_\lambda$) $= 0$ for any Σ_λ received from $P_\lambda \in \mathcal{R}$ where $\mathcal{R}$ is the set of signers who send their signatures to P_i.

Security Definition

The formal definition is given below.

- **Setup**. $\mathcal{C}$ runs Setup(1^k) to obtain the public parameter *pp*. $\mathcal{C}$ runs Keygen(*pp*) to generate a challenge key-pair $(pk^\star, sk^\star)$. $\mathcal{C}$ initializes the list of certified public keys $C \leftarrow \varepsilon$, an runs algorithm $\mathcal{A}$ with $pk^\star$ as its input.
- **Certificate Queries**. $\mathcal{A}$ provides a key pair (pk, sk) in order to certify *pk*. $\mathcal{C}$ adds *pk* to C if *sk* is its matching private key.
- **PSigning Queries**. When $\mathcal{A}$ requests a provenance signature under $pk^\star = pk_i$, with loc and $\{\Sigma_\lambda\}$ where $\Sigma_\lambda = (G_\lambda, \sigma_\lambda)$ from $P_\lambda \in \mathcal{R} \subset \mathcal{S}$ and σ_λ is a provenance signature for G_λ generated by signer P_λ, this query is answered with a provenance signature $\Sigma = (G, \sigma)$ where the corresponding identity $\mathrm{id}^\star$ of $pk^\star$ is encoded in G, or $\perp$ if any of the input $\{\sigma_\lambda\}$ is invalid.
- **Output.** Eventually, $\mathcal{A}$ outputs $\Sigma = (G^*, \sigma^*)$, which is a valid forgery if
  1. $\mathrm{PVrf}(pp, \Sigma) = 1$.
  2. $pk^\star = pk_{i^*}$, for some $i^* \in \{1, \cdots, N\}$ with the corresponding identity encoded in G^*.
  3. All public keys whose identities are encoded in G^* (except the challenge key pk_{i^*}) are encoded in C.
  4. $\mathcal{A}$ has never queried any G' that contains pk_{i^*} with G' being a subgraph of G^*.

The advantage of $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{A}}$, is the probability that it wins the above game, where the probability is taken over the coins of Setup, KeyGen and $\mathcal{A}$ itself. In the random oracle model, the probability is also over the choice of the random function(s) implemented by the random oracle(s).

Definition (security)

We say that $\mathcal{A}$ $(\mathrm{T}, q_p, \epsilon)$-breaks the provenance signature scheme if it runs in time at most T, makes at most q_p signature queries to the **PSigning** oracle, and has an advantage $\mathrm{Adv}_{\mathcal{A}}$ of at least ϵ. If there is no such an adversary, we say that the provenance signature scheme is $(\mathrm{T}, q_p, \epsilon)$-secure under a chosen message attack.

Our Construction

We use the BLS signature as the building block.

- Setup(1^k): Generate a bilinear group $\mathbb{G}$ with order $2^{k+1} \geq p \geq 2^k$ and an associated bilinear pair $e(\cdot, \cdot) : \mathbb{G} \times \mathbb{G} :\longrightarrow \mathbb{G}_\mathsf{T}$. Return $pp = (e, \mathbb{G}, \mathbb{G}_\mathsf{T}, H)$, where $H : \{0,1\}^* \rightarrow \mathbb{G}$ a random oracle.
- Keygen(pp): Randomly choose $x \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and output a pair of private and public keys ($sk = x, pk = X = g^x$).
- GraphCom($pp, \mathrm{loc}, \{G_\lambda\}_{\lambda \in \mathcal{R}}$): On input of public parameters pp, local information string $\mathrm{loc} = (\mathrm{id}_i, m_i, t_i)$ for a local message m_i of trustworthiness t_i, and incoming $|\mathcal{R}|$ provenance subgraphs $\{G_\lambda\}_{\lambda \in \mathcal{R}}$ where $\mathcal{R} \subset \mathcal{S}$, P_i generates a new message $\overline{m}_i = \mathrm{alg}_i(m_i, \{G_\lambda\}_{\lambda \in \mathcal{R}})$ of trustworthiness $\bar{t}_i = \mathrm{tru}_i(t_i, \{G_\lambda\}_{\lambda \in \mathcal{R}})$, where the specification of algorithms alg_i and tru_i is application-dependent and beyond the scope of the paper. Finally, P_i outputs a provenance subgraph $G_i = ((\{G_\lambda\}_{\lambda \in \mathcal{R}}), a_i)$ for its newly produced message $\bar{m}_i$, where $a_i = (\mathrm{id}_i, \mathrm{alg}_i, \overline{m}_i, m_i, \mathrm{tru}_i, \bar{t}_i, t_i)$ is the "end" node in G_i. Note that if $\{G_\lambda\} = \emptyset$, then $((G_\lambda), a_i) = (a_i)$.

- PSign($pp, sk_i, \mathrm{loc}, \{\Sigma_\lambda\}_{\lambda\in\mathcal{R}}$): on input of public parameters pp, local information $\mathrm{loc} = (\mathrm{id}_i, m_i, t_i)$ of message m_i of trustworthiness t_i, a private key sk_i of P_i, and provenance signatures $\{\Sigma_\lambda\}_{\lambda\in\mathcal{R}}$ on respective provenance subgraphs $\{G_\lambda\}_{\lambda\in\mathcal{R}}$ received from P_i's upstream nodes belonging to $\mathcal{R} \subset \mathcal{S}$, P_i executes as follows:
  1. Execute PVrf(pp, Σ_λ), which is specified below, to verify the individual provenance signatures Σ_λ. If any verification fails, abort.
  2. Set $G_i \leftarrow \mathrm{GraphCom}(pp, \mathrm{loc}, \{G_\lambda\}_{\lambda\in\mathcal{R}})$
  3. Use algorithm $\mathrm{BSig}_{sk_i}(\cdot)$ to obtain $\omega \leftarrow H(G_i)^{x_i}$
  4. Output $\Sigma_i = (G_i, \sigma_i)$ where $\sigma_i = \omega \prod_{P_\lambda\in\mathcal{R}} \sigma_\lambda$.
- PVrf(pp, Σ): given parameters pp, provenance signature $\Sigma = (G, \sigma)$, the algorithm parses G to obtain $\{G_i | i = 1, \cdots, \ell\}$ and the signers' identities $\{\mathrm{id}_i | i = 1, \cdots, \ell\}$, and returns 1 if the following equation holds and 0 otherwise:

$$e(g, \sigma) \stackrel{?}{=} \prod_{i=1}^{\ell} e\left(X_i, H(G_i)\right).$$

Security Result

Theorem

If the $\mathcal{BLS}$ signature is $(\tau', q_s, \varepsilon')$-secure under a chosen-message attack, our provenance signature scheme is (τ, q_p, ε)-secure where

$$\varepsilon \geq \varepsilon', \quad q_p = q_s \quad \textit{and} \quad \tau \leq \tau' - (q_s + 1)N \cdot \tau_e, \tag{1}$$

where q_s, q_p are the numbers of the queries to the $\mathcal{BLS}$ signing oracle and the **PSigning** *oracle, respectively, and τ_e is the time cost of exponentiation computation.*

Optimal Security Hardening

The fact that hardening security is often costly naturally leads to the problem of optimal hardening — an optimization problem. Specifically, given a trustworthiness graph, we want to identify the most “influential” K nodes so as to harden their security.

Theorem

The optimal security hardening problem for trustworthiness graphs is NP-hard.

We now show that the optimal security hardening problem also has a certain submodular structure, and thus the problem renders to some natural greedy algorithm that can produce solutions within a constant approximation factor of the optimal solution. A function $f(\cdot)$ mapping sets to $\mathbb{R}^+$ is said to be submodular if it has the so-called *diminishing returns* property: for all $v \in V$ and all $A \subseteq B$ it holds that

$$f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B).$$

By defining $\sigma(A)$ as the expected number of nodes "influenced" by the nodes in $A \subseteq V$ (i.e., the expected number of principals that accept the malicious information inserted into the information network by the corrupt principals belonging to A) and the following theorem

Theorem (Nemhauser78)

For a non-negative, monotone submodular function f, let S be a set of size K obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let S^* be a set that maximizes the value of f over all K-element sets. Then $f(S) \geq (1 - 1/e) \cdot f(S^*)$.

We show that the optimal security hardening problem is submodular.

Theorem

The function $\sigma(\cdot)$ incurred by the optimal hardening problem is submodular.

Heuristic Algorithms for Solutions

- Greedy:At each step with an already selected node set A, which is initially empty, we select v that leads to maximal $\sigma(A \cup \{v\}) - \sigma(A)$.
- Random: At each step, we uniformly select a yet-to-be-selected node at random.
- Heuristic: Given $G = (V, E)$, we select the K highest out-degree nodes.
- Heuristic+: At each step, we select the highest out-degree node in the graph that is obtained after deleting the nodes that have been selected or "influenced", and their outgoing and incoming arcs. This algorithm can be seen as a hybrid of the above Greedy algorithm and Heuristic algorithm.

Results

The Greedy and Heuristic+ algorithm are more effective (with the latter being $O(|V|)$ faster)

For specific results and simulation, please refer to the paper.

Related Work

- Inspired by our earlier related framework for "trustworthiness-centric information sharing" [IFIPTM'09].
- Different from "information flow" (trustworthiness $>$secrecy + integrity); e.g., what if a bad guy inserts malicious information into a system?
- Our network-level differs from OS/DB-level because we allow compromised OS/DB.
- Our provenance signatures move a step beyond recent similar proposals [Hasan et al. FAST'09; Zhang et al. VLDB-SDM'09]:
  - Better security: no peeling off attack because of aggregation
  - Better efficiency: no linear increasing in signature size

Conclusions

Our Results

1. We present the concept of “information trustworthiness management” in the context of information networks and abstract “trustworthiness graph”.
2. We identify a new kind of cryptographic primitive, “provenance digital signatures” preserving the history of a message.
3. We analyze the optimal security hardening and show that the problem in question is NP-hard, but has a good approximation algorithm.

There are many interesting problems for future investigations.

Further Work

1. A first important issue is to efficiently maintain the trustworthiness graphs that in most cases dynamically evolve with time.
2. Another important issue is represented by suitable abstractions that can serve as a base for modeling, reasoning, discussing information trustworthiness.
3. The relationships of our mechanisms with access control mechanisms and privacy also need to be investigated.

THANK YOU.