Br2Br: A Vector-based Trust Framework for WebRTC Calling Services

Ibrahim Tariq Javed, Khalifa Toumi, Noel Crespi, Amir Mohammadinejad
Institut Mines-Telecom
Telecom SudParis, Evry, France
Email: {Ibrahim_Tariq.Javed, Khalifa.Toumi, Noel.Crespi, Amir.Mohammadinejad@telecom-sudparis.eu}

***Abstract*—WebRTC provides web calling services by enabling communication between browsers in a P2P fashion. To achieve security and enhance user privacy it is essential to define the level of trust between the various entities involved in WebRTC security architecture. The existing P2P trust models are not directly applicable as the characteristics of browser to browser communication are not taken into account. We, therefore present 'Br2Br' a vector based trust framework for defining trust in WebRTC. The concepts of trust requirement, trust context, trust policy and trust evaluation are proposed to define trust relationships in WebRTC architecture. The framework considers identification, reputation and experience parameters to evaluate the amount of trust, distrust and mistrust. Finally we present set of characteristics and behaviors essential for the evaluation of trust in WebRTC communications.**

Keywords*—*WebRTC; Trust; Identity; IdP; Authentication; Reputation.

I. Introduction

The advancements in web communication services and in HTML technology has prompted the development of WebRTC standard [1], which has brought P2P real time capabilities to web browsers for the very first time and without the use of any plug-in. WebRTC enables the distribution of P2P functionality as web applications that run distributively in web browsers downloaded from a central server. This open source web technology allows web pages to have calling feature with just a few lines of code [2] whereas any device that can run a WebRTC enabled browser can now access real time communication services ubiquitously. Therefore WebRTC is envisioned to set the stage for an explosion of context-based web calling services.

The WebRTC security architecture decouples authentication from the Calling Server (CS), allowing communicating participants to independently validate each other using their third party Identity Providers (IdPs) [3]. Before establishing connection the browser is required to validate the identity of the remote entity. The origin of service providers are validated using the certificate based Public Key Infrastructure (PKI), whereas the identities of communicating participants are authenticated in a P2P fashion [4]. Nevertheless, trust can not be established solely on the basis of authentication. It needs to be evaluated in an efficient and reliable manner.

Recognizing the importance of trust in browser to browser communications, the immediate question is how to define and establish trust between various entities involved. For instance partially and fully trusted models for identity provisioning in WebRTC have been presented and their impact on user privacy examined [5]. Security improvements and mitigating techniques for the endpoint authenticity are proposed in [6]. Whereas in [7], it is the relationships of users with their IdP and CS that are inspected in order to provide new trust requirements for WebRTC security architecture. However, no attempts were made to evaluate trust between entities of WebRTC. Certain P2P trust models exist [8]–[10] but none of them takes into account the characteristics of browser to browser communication.

Several solutions [11], [12] exist for the WebRTC security challenges identified in [13] but none of them evaluates the dynamic and changing behavior of entities in terms of user security. Therefore we present "Br2Br", a vector-based trust framework for WebRTC security architecture to incorporate the dynamic nature of entities. To the best of our knowledge, this is the first effort to formalize a generic model for trust in WebRTC. We believe that our model will help evaluate trust to overcome uncertainty and risk in browser based web calling. The evaluated trust may be used to enhance security and privacy by implementing privacy preservation techniques [14] and policy decisions for browsers.

This framework formalizes the three trust relationships of WebRTC: User-IdP, User-CS and User-User. To define these relationships the concepts of trust evaluation, trust policy, trust context and parameters influencing trust are presented. Our framework formalizes the dependence of trust on time and on a particular context where the evaluation depends upon three parameters: experience, reputation and identification. The notion of different degrees of trust are introduced, differentiating between trust, distrust and mistrust adopted from Jϕsang's opinion model [15]. Finally based on the security and trust requirements of WebRTC, various behaviors and identification characteristics are presented.

The rest of this paper is structured as follows: Section II describes the WebRTC security architecture. The vector based trust framework is presented in Section III and the parameters used for trust evaluation are formalized in Section IV. The set of behaviors and characteristics considered for evaluating trust are presented in Section V. Section VI presents a user scenario utilizing Br2Br framework. Finally, in Section VII we provide our conclusion.

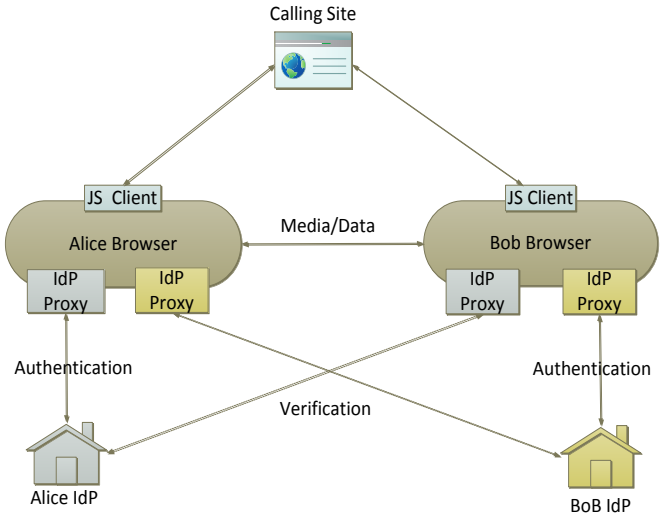

Fig. 1. WebRTC Security Architecture

II. WebRTC Security Architecture

WebRTC standard is an open source web technology that provides real time communication capabilities to browsers via simple APIs. It is envisioned to allow existing telecom operators and OTT players the incentive of having free, open, global and inter-operable communication flows over the web [16]. A new communication framework using the underlying WebRTC technology is being developed [17] whereas 3GPP offers the interconnection of WebRTC with IMS [18]. WebRTC is expected to bring a wide range of possibilities for corporate and personal communications over the web.

In WebRTC, the calling site is a web server that enables communicating participants to exchange information by providing JS client that executes on the browser. The CS is responsible for providing signaling between the two parties for the exchange of session parameters, identities, call answer/offer request and user reachable addresses. WebRTC aims in having minimum level of trust in CS by decoupling the authentication procedures from the signaling. Authentication of communicating participants is managed by service-independent IdP [19] using existing Single Sign On protocols such as OAuth2.0, OpenID Connect, SAML etc.

Figure 1 presents WebRTC security architecture [20] in an Alice-Bob call scenario. The CS provides a calling interface for Alice to discover Bob and initiate a call request. To authenticate Alice, Alice's browser downloads an IdP Proxy from Alice's IdP. Upon successful authentication, the IdP server returns an identity assertion containing Alice's identity information. The assertion is attached to the call request sent to Bob via the CS. When Bob receives the call request, Bob's browser instantiates Alice's IdP Proxy and passes on this assertion in order to verify Alice's Identity. Upon successful verification the authentication result is shown to Bob.

In WebRTC users trust their calling services to connect them to authorized parties and treat their personal data and accumulated call history confidential. On the other hand IdPs are trusted to store and manage their personal profile information in a secure and efficient manner while preserving their privacy [21]. However, users trust their communication participants to access media/data streams based on the level of identification they provide. In WebRTC, web browser is the only entity that user trusts completely. Therefore it initiates the authentication process for each entity on behalf of the user. However, trust cannot established by merely validating the identities of each entity [22]. An efficient trust management system to estimate the trustworthiness of communicating participants and the service provides is essential.

III. Br2Br Vector based Trust Framework

We introduce the concept of trust in order to manage the security of information exchanged in WebRTC services by proposing a new trust framework "Br2Br". Figure 2 illustrates the basic concept of the trust framework, which includes three types of trust relationships: User-CS, User-User and User-IdP. In our model trust is influenced by three parameters: experience, identification and reputation.

In this framework, entities are characterized into one of the two types, either a truster, the entity which establishes trust, or a trustee, the entity which is being trusted. A user's browser is the only entity that is considered as a truster, whereas CSs, IdPs and the communicating participants browsers are considered as trustees. We represent trust as a relation between user U and an entity E within a context c at time $\dot{t}$ such that:

$$(U \xrightarrow{c} E)_{\dot{t}} \tag{1}$$

where time $\dot{t}$ is used to characterize the dynamic behavior of a trust relationship over a specific time period $[t_0, t_n]$. The time period is divided into n subintervals $[t_0, t_1], [t_1, t_2], ..., [t_{n-1}, t_n]$. The k^{th} interval is represented

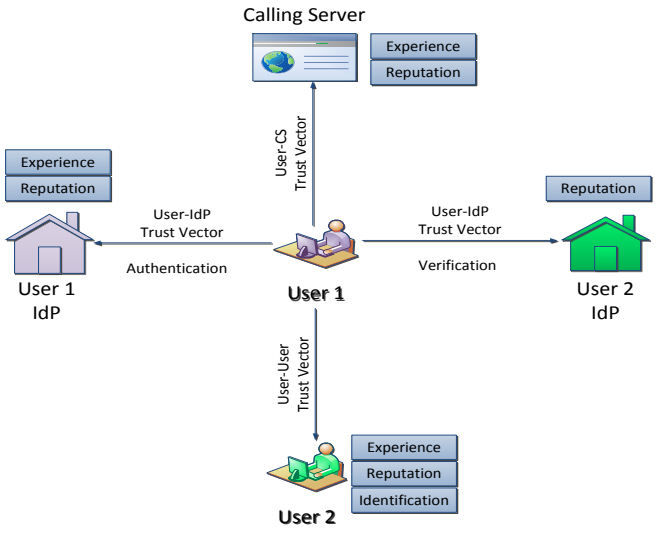

Fig. 2. WebRTC Trust Model

as $[t_{k-1}, t_k]$ where $k = 1, 2, ..., n$. The context c is the information that characterizes the situation of entities involved. The notion of context is defined by combining the concepts of trust objectives and trustee aspects.

Definition 1: Trust objective is the purpose to form a trust relationship, whereas aspects are the characteristics oof trustee considered by the truster. Therefore, let CONTEXT, OBJECTIVE and ASPECTS be the set of all possible contexts, objectives and aspects respectively, where each $c \in$ CONTEXT is a tuple (o, a) and where $o \in$ OBJECTIVES and a $\in$ ASPECTS.

We consider two types of trust objectives in WebRTC: to access resources and to provide services. Services include communication and authentication whereas resources include media stream and identity assertions. The aspects of trustee considered are security, reliability, confidentiality and honesty. Therefore the trust relationship in WebRTC is never absolute. A truster will always trust a trustee with respect to the set of specific objectives and aspects defined by the trust context. For example, user U trusts trustee E's security and confidentiality to provide authentication.

Example 1: Alice uses a web calling site "example.com" to place calls from her browser. She trusts the CS to provide communication services in a secure and reliable manner. This does not mean that the CS will also be trusted to access Alice's identity information and media streams. Meanwhile, Alice trusts her friend Bob to access her identity information and media streams in a confidential manner.

In our model, trust is represented in the form of a trust triple (t, d, m), where t represents trust, d represents distrust and m represents mistrust. Unlike single trust values this vector representation of trust allows us to show the amount of trust, distrust and uncertainty within each WebRTC relationship.

Definition 2: We represent trust using a trust triple (t, d, m) where $t, d, m \in [0, 1]$ and $t + d + m = 1$. Trust t is the expectation that an entity will perform reliably, securely and confidentially within a specific context. Distrust d is the expectation that an entity will not perform reliably, securely and confidentially within a specific context and Mistrust m is a level of doubt that an entity will perform reliably, securely and confidentially within a specific context.

The trust relation is a 3×3 matrix. The rows of the matrix correspond to three parameters, experience, recommendation and identification. The formal definition and evaluation of each parameter is provided in Section IV. Each of these parameters are represented in the rows of a trust matrix, where each term of the trust triple represents the columns of trust matrix.

$$\begin{pmatrix} t^E & d^E & m^E \\ t^R & d^R & m^R \\ t^I & d^I & m^I \end{pmatrix} \tag{2}$$

The three parameters may not be of equal importance in evaluating trust. For example, a truster U may place more significance on the identification parameter rather than experience and reputation. Therefore, we present a weight scheme vector that specifies the relative weights for each parameter to evaluate trust triples. The user's trust evaluation policy will define the weight scheme vector.

Definition 3: The weight scheme is a vector of the form $(S^E, S^R, S^I)_{U \to E}$. The elements of vector are the weights assigned to the parameters in the trust matrix such that $S^E + S^R + S^I = 1$ and $S^E, S^R, S^I \in [0, 1]$.

U's trust on E within a specific context c is thus represented by a single trust triple, as follows:

$$(t^c, d^c, m^c)_{U \to E} = (S^E, S^R, S^I) \times \begin{pmatrix} t^E & d^E & m^E \\ t^R & d^R & m^R \\ t^I & d^I & m^I \end{pmatrix} \tag{3}$$

where $t^c = S^E \times t^E + S^R \times t^R + S^I \times t^I$, $d^c = S^E \times d^E + S^R \times d^R + S^I \times d^I$ and$m^c = S^E \times m^E + S^R \times m^R + S^I \times m^I$

However the trust relationship should not only depend on the current values evaluated, it should also depend on the old values of trust. For example, if truster U completely trusts the trustee E then negative factors will be often overlooked when trust is re-evaluated. Therefore we present the final trust vector at time $\dot{t}$ as a linear combination of the previous time-dependent trust (t_i, d_i, m_i) and the trust evaluated at the present time (t^c, d^c, m^c). The weights assigned to old and current trust vectors is a matter of a user's trust evaluation policy.

Definition 4: To evaluate the final trust vector the relative weight α is assigned to the trust obtained at the present time and $1 - \alpha$ to the previous time-dependent trust vector, where $\alpha \in [0, 1]$.

Thus the final trust evaluated between a truster U and trustee E at time $\dot{t}$ in a particular context c is defined as:

$$\begin{aligned} (U \xrightarrow{c} E)_{\dot{t}} &= \alpha \times (t_i, d_i, U_i) + (1 - \alpha) \times (t^c, d^c, m^c)) \\ &= ({}_U t^c_E, {}_U d^c_E, {}_U m^c_E) \end{aligned} \tag{4}$$

where ${}_U t^c_E = \alpha \times t_i + (1-\alpha) \times t^c$, ${}_U d^c_E = \alpha \times d_i + (1-\alpha) \times d^c$ and ${}_U m^c_E = \alpha \times m_i + (1 - \alpha) \times m^c$

IV. Trust Model Parameters

In this section we formally define the three parameters, experience, reputation and identification, along with their respective evaluation. The Br2Br framework is easily extendable for the inclusion of other parameters, such as Knowledge.

A. Experience

The experience parameter is based on the past performance of the trustee in the given context [23]. In our trust model the performance is evaluated based on the behavior of trustee. We consider four types of behaviors encountered by the truster: good, bad, neutral and undisclosed.

Definition 5: Experience parameter (t^E, d^E, m^E) is defined as the computation of the aggregate performance of a trustee based on its behavior detected in a particular context over a specified period of time .

We model experience in terms of the number of behaviors encountered by a truster in a context over n subintervals of

time period $[t_0, t_n]$. Let G_k, B_k, N_k, U_k be set of all good, bad, neutral and undisclosed behaviors that occur in the k^{th} interval $[t_{k-1}, t_k]$ of the time period. The experience acquired in the k^{th} interval is represented by (t_k, d_k, m_k) and evaluated as follows:

$$t_k = \frac{|G_k| + |\frac{N_k}{2}|}{|G_k| + |B_k| + |N_k| + |U_k|}$$

$$d_k = \frac{|B_k| + |\frac{N_k}{2}|}{|G_k| + |B_k| + |N_k| + |U_k|}$$

$$m_k = \begin{cases} 1 & \text{if} G_k = B_k = N_k = U_k = 0 \\ \frac{|U_k|}{|G_k|+|B_k|+|N_k|+|U_k|} & otherwise \end{cases} \quad (5)$$

The intuition behind the evaluation of experience is that each good, bad and undisclosed behavior contributes to the trust, distrust and mistrust components respectively by a factor of $\frac{1}{|G_k|+|B_k|+|N_k|+|U_k|}$, whereas, the neutral behavior contributes to both trust and distrust components by a factor of $\frac{0.5}{|G_k|+|B_k|+|N_k|+|U_k|}$. However, if no behavior occurs in k^{th} time interval then the mistrust component is equal to 1 and $t_k = d_k = 0$.

Naturally, the behaviors that occur in the older intervals should be weighted less than the behaviors in recent intervals. Each interval $[t_{k-1}, t_k]$ is thus weighted based on its position. We use the position weight p_k for each interval calculated, using $p_k = \frac{k}{S}$ where $S = \frac{n(n+1)}{2}$ [24]. Therefore the experience parameter is evaluated as $t^E = \sum_{i=1}^{n} p_k \times t_k$, $d^E = \sum_{i=1}^{n} p_k \times d_k$ and $m^E = \sum_{i=1}^{n} p_k \times m_k$.

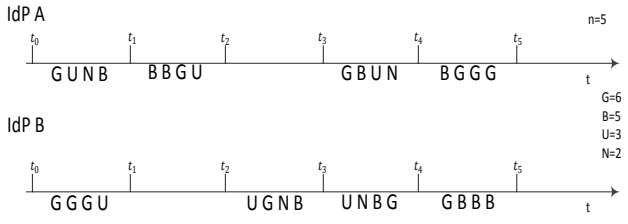

Fig. 3. Set of Behaviors

Example 2: Bob uses the services of two different IdP's to authenticate himself over various web calling sites. To put trust in IdP he only considers the experience parameter. Figure 3 shows the set of IdP A and IdP B behaviors that Bob has experienced over a time period $[t_0, t_5]$ where $n = 5$. The position weights assigned to each interval are $p_1 = \frac{1}{15}, p_2 = \frac{2}{15}, p_3 = \frac{3}{15}, p_4 = \frac{4}{15}, p_5 = \frac{5}{15}$. Both sets have the same number of good, bad, neutral and undetermined behavior. However, the trust triple for IdP A is (0.4,0.28,0.32) whereas for IdP B it is (0.31,0.42,0.27). IdP A has a higher level of trust value only because it has more good behaviors that have occurred more recently than those of IdP B.

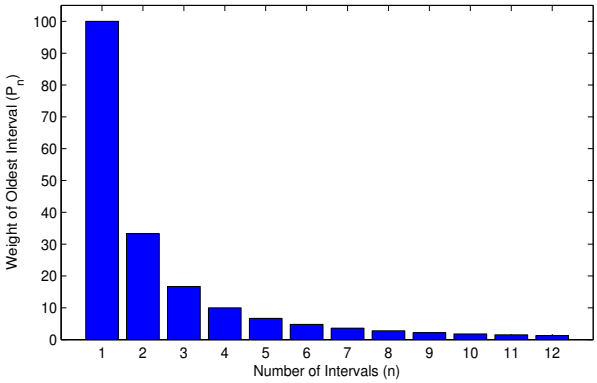

Fig. 4. Oldest Interval Impact

The experience parameter not only depends upon the weight-age of each interval but also on the total number of intervals n considered by the truster. Figure 4 represents the weightage of the oldest interval with varying n. It can be observed that as the number of intervals increases, the weight of the oldest interval gets so small that it has no significant impact on the current value of trust. The selection of total number of intervals for computing the experience is again a matter of user's trust evaluation policy. Users may choose to forget the behaviors that are older than a particular amount of time. However, the decision should depend on the requirement of the accuracy of trust and the storage cost per interval.

B. Identification

The Identification parameter measures the amount of trust that a user can place in a digital identity received to authenticate the communicating participant. Several characteristics of the identity assertion are considered, wherein each characteristic consists of various identification levels. The identification levels are provided by the IdP during the identity verification process.

Definition 6: The identification parameter (t^I, d^I, m^I) determines the strength in the authentication process of the communicating participant. It is the aggregate of all satisfactory, unsatisfactory and unproven identification levels of the digital identity transaction weighted with the amount of trust in the IdP providing the authentication information.

The characteristics are represented by alphabets such as "X" and consists of various identification levels such as $X0, X1, X2, X3....etc$ further explained in Section V. Each level is considered to be satisfactory, unsatisfactory or unproven attribute of the identity assertion. This categorization of identification levels are based on user trust evaluation policy.

Let IdP 'i' be the entity that provides the authentication information for the communicating participant p to user U. Where as Sat, $Unsat$ and $Unprov$ are the set of satisfactory, unsatisfactory and unproven identification levels considered by user U. Then the identity trust triple (t_p, d_p, m_p) for the communicating participant p is defined as the average aggregate of

the number of satisfactory, unsatisfactory and unproven identification levels such that $t_p = \frac{|Sat|}{|Sat|+|Unsat|+|Unprov|}$, $d_p = \frac{|Unsat|}{|Sat|+|Unsat|+|Unprov|}$ and $m_p = \frac{|Unprov|}{|Sat|+|Unsat|+|Unprov|}$.

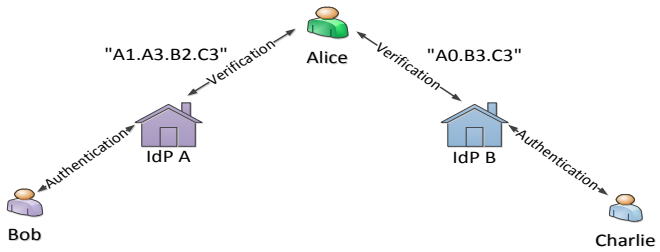

Fig. 5. Identification Scenario

Example 3: Alice communicates with Bob and Charlie who are authenticated from IdP A and IdP B, respectively, as shown in Figure 5. Three characteristics "A", "B" and "C" are considered to evaluate the strength in the identity assertion. During identity verification, the IdPs provide the identification levels as "A1.A3.B2.C3" and "A0.B3.C3" for Bob and Charlie respectively. The trust evaluation policy of Alice considers the sets $Sat = \{A2, B2, B3, C3\}$ $Unsat = \{A1, A3, B1, C1, C2\}$ and $Unprov = \{A0, B0, C0\}$. Using the formulas of t_p, d_p and m_p the identity trust triples for Bob and Charlie are calculated to be $(\frac{2}{4}, \frac{2}{4}, 0)$ and $(\frac{2}{3}, 0, \frac{1}{3})$ respectively.

However to evaluate the identification parameter the amount of trust in the IdP providing the authentication information should also be taken into account. Therefore if the trust triple between user U and IdP i is denoted by $({}_Ut_i, {}_Ud_i, {}_Um_i)$ then the identification trust triple (t^I, d^I, m^I) is evaluated by weighting it with the user's trust in the IdP :

$$\begin{aligned} t^I &= {}_Ut_i \times t_p \\ d^I &= {}_Ut_i \times d_p \\ m^I &= {}_Ud_i + {}_Um_i + {}_Ut_i \times m_p \end{aligned} \tag{6}$$

The intuition behind the weightage assessment is that the user considers the authentication information trustworthy only if that user trusts the IdP otherwise it ignores the information making the mistrust factor of the identification parameter even higher.

Example 4: From the previous example it seems that Alice will put more trust in the authentication process of Charlie compared to that of Bob. However, this may not necessarily be the case. Let us suppose that the triple for Alice's trust in IdP A is $(0.9, 0.1, 0)$ whereas the triple for Alice's trust in IdP B is $(0.1, 0.7, 0.2)$. Using Equation 6, the identification parameters (t^I, d^I, m^I) for Bob and Charlie are evaluated to be $(0.45, 0.45, 0.1)$, and $(0.067, 0, 0.933)$, respectively making Bob's authentication more trustworthy. This is due to the fact that Alice ignores IdP B's authentication information about Charlie increasing the uncertainty in Charlie's identification.

C. Reputation

The reputation parameter aggregates the endorsements received about an entity from user's various communicating participants. An endorsement about an entity E is a trust triple $({}_pt_E, {}_pd_E, {}_pm_E)$ provided to the user by a communicating participant p. However, each endorsement should be weighted with the amount of trust in the communicating participant. Therefore we consider reputation to be collective measure of the endorsements from members of a particular community where each community is weighted according to the trust of the user in that community.

Definition 7: Reputation parameter (t^R, d^R, m^R) is the weighted aggregate of the average endorsements about a trustee received by each communicating participant of a particular community in a specific context.

We define 7 levels for endorsements in Table I, where each endorsement level corresponds to a specific trust triple $({}_pt_E, {}_pd_E, {}_pm_E)$ provided to the user U by a communicating participant p about an entity E. However each participant belongs to a particular community of the user's contact list such as friends, classmates, relatives, co-workers etc. Therefore we consider the aggregate community trust triple $({}_ct_E, {}_cd_E, {}_cm_E)$ as the average of all endorsements received by the communicating participants of the community c such that ${}_ct_E = (\frac{\sum_{i=1}^{\bar{n}} {}_{e_i}t_E}{\bar{n}})$, ${}_cd_E = (\frac{\sum_{i=1}^{\bar{n}} {}_{e_i}d_E}{\bar{n}})$ and ${}_cm_E = (\frac{\sum_{i=1}^{\bar{n}} {}_{e_i}m_E}{\bar{n}})$ where $\bar{n}$ are the total number of endorsers in the community. However, each community trust triple should be weighted with the amount of user trust in that community.

Definition 8: Let $\hat{n}$ be the total number of communities set by the user, then the corresponding community weight vector is $(W_{c_1}, W_{c_2}, \cdots, W_{c_{\hat{n}}})$ such that $(W_{c_1} + W_{c_2} + \cdots + W_{c_{\hat{n}}}) = 1$ and $W_{c_1}, W_{c_2}, \cdots, W_{c_{\hat{n}}} \in [0, 1]$.

Therefore the community trust matrix consists of $\hat{n}$ rows where each row correspond to the trust triple of a particular community. The reputation parameter (t^R, d^R, m^R) is a multiplication of the community trust matrix and the corresponding community weight vector of the user:

$$(t^R, d^R, m^R) = (W_{c_1} W_{c_2} \cdots W_{c_{\hat{n}}}) \times \begin{pmatrix} {}_{c_1}t_E & {}_{c_1}d_E & {}_{c_1}m_E \\ {}_{c_2}t_E & {}_{c_2}d_E & {}_{c_2}m_E \\ \vdots & \vdots & \vdots \\ {}_{c_{\hat{n}}}t_E & {}_{c_{\hat{n}}}d_E & {}_{c_{\hat{n}}}m_E \end{pmatrix} \tag{7}$$

where

$$\begin{aligned} t^R &= W_{c_1} \times {}_{c_1}t_E + W_{c_2} \times {}_{c_2}t_E + \cdots + W_{c_n} \times {}_{c_n}t_E \\ d^R &= W_{c_1} \times {}_{c_1}d_E + W_{c_2} \times {}_{c_2}d_E + \cdots + W_{c_n} \times {}_{c_n}d_E \\ m^R &= W_{c_1} \times {}_{c_1}m_E + W_{c_2} \times {}_{c_2}m_E + \cdots + W_{c_n} \times {}_{c_n}m_E \end{aligned}$$

Example 5: Alice has set up two communities in her contact list $(family, friends)$. The corresponding community weight vector is $(0.8, 0.2)$. Alice usually avoids picking up calls that

TABLE I
ENDORSEMENT LEVELS

Endorsement Levels	Trust triple
Uncertain	$(0, 0, 1)$
Trusts absolutely	$(1, 0, 0)$
Trusts moderately	$(\frac{3}{4}, \frac{1}{4}, 0)$
Trusts neutrally	$(\frac{1}{2}, \frac{1}{2}, 0)$
Distrusts moderately	$(\frac{1}{4}, \frac{3}{4}, 0)$
Distrusts absolutely	$(0, 1, 0)$
No response	$(0, 0, 1)$

are not in her contact list, however before rejecting a call request she considers caller's reputation. If the trust value t^I of the caller is very high she accepts the call because that makes her feel that the person calling is very well known and trusted by her family members.

V. TRUST RELATIONSHIPS

In Br2Br, each relationship is represented by a trust vector. The evaluation of trust vector is based on the 1) context; 2) trust policy; 3) type of trustee; 4) appropriate parameters. The User-CS and User-IdP trust vectors are based on the experience and reputation parameters, whereas the User-User trust vector also considers the identification parameter. The experience in Br2Br is based on the past behavior of the entity whereas identification depends on the characteristics of the identity assertion. In this Section, we present set of behaviors and identity characteristics for the evaluation of experience and identification parameters in WebRTC communications.

A. User-User Trust

The trust context in a User-User trust relationship is defined as the user's trust in the communicating participant's security and honesty to access identity information and media streams. Using the Electronic Authentication Guideline [25] we present three characteristics of identity assertion: Identity Proofing, Credential Strength and Assertion Endurance, to estimate the trustworthiness of communicating participants. Each characteristic is further represented by different identification levels. These levels can be used to evaluate trust triple or can be indicated in plain text/symbols to the user.

Identity Proofing: This characteristic defines how strongly the set of identity information representing a person has been verified by the IdP. This characteristic is represented by the following levels:

- $P0$ No information about proofing is provided by the IdP;
- $P1$ A pseudonymous identity is used;
- $P2$ Identity information is self proclaimed;
- $P3$ Identity information is proofed using social proofing;
- $P4$ Identity information is proofed using signed/notarized documents;
- $P5$ Identity information is proofed in person.

Credential Strength: This characteristic defines how strong user credentials are and how easily they can be spoofed or stolen. The characteristic is represented by the following levels:

- $C0$ No information about credentials is provided by the IdP;
- $C1$ No credentials are used;
- $C2$ Credentials having username/password combination;
- $C3$ Shared secret using symmetric key encryption;
- $C4$ Cryptographic proof using asymmetric key;
- $C5$ Hard tokens employed using trusted biometrics.

Assertion Endurance: This characteristic shows how well the identity assertion is protected against unauthorized access. The characteristic is represented by the following levels:

- $S0$ No information about assertion is provided by the IdP;
- $S1$ The identity assertion is neither protected nor signed;
- $S2$ An access token is used to retrieve identity assertion;
- $S3$ Identity assertion is signed and verifiable by the IdP;
- $S4$ Identity assertion is encrypted;
- $S5$ Identity assertion is audience protected.

Example 6: A bank provides remote financial assistance using WebRTC calling server. The bank requires customers to authenticate from a set of trusted IdPs. However, to provide security and confidentiality the bank representative limits financial information based on the strength of customers identification. Let's suppose $Customer1$ and $Customer2$ have identification levels as "$P5.C4.S3.S4.S5$" and "$P2.C2.S1$" respectively. Due to strong identification characteristics the bank representative allows $Customer1$ to receive sensitive information regarding personal account transactions. However, it restricts $Customer2$ to only obtain general information about bank services due to fragile identification.

B. User-CS Trust

The trust context in User-CS relationship is the user's trust of a CS's security and reliability to provide communication services. Utilizing the well established WebRTC security requirements [26] we provide a set of behaviors that should be considered to evaluate the experience parameter for web calling services.

Mixed Content: In WebRTC, user interconnection with CS is considered to be secure if data is transferred over HTTPS [20]. However, the CS may produce mixed content during the duration of the call by loading JS from an HTTP origin over its HTTPS page. The JS from HTTP might redirect media to location controlled by the attacker.

IdP Selection: Current WebRTC specifications allow a CS to enforce the selection of a particular IdP. If the *setIdentityProvider* method has been called by the CS, then the user is bound to authenticate from a particular IdP [1] set by the CS. This may lead to privacy and security concerns as a user may not trust the IdP to which it is forced to authenticate.

JS Client Load Time: This indicates the time in seconds required to receive all the elements from the CS while loading the JS client. The user will only be able to place or receive calls from the browser on successful loading of JS client. The reliability of CS will depend on the time it takes for loading the JS client to be loaded on to the browser.

Response Waiting Delay: This delay specifies the time in seconds spent by the browser waiting for a response message

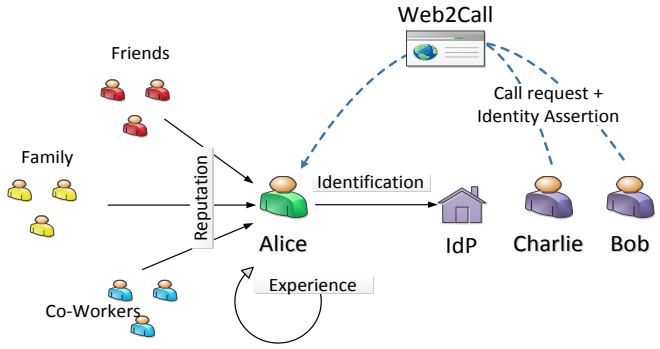

Fig. 6. User Scenario Example

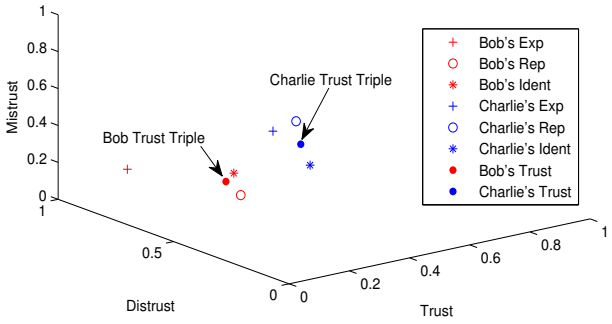

Fig. 7. Bob and Charlie Trust Representation

from the server. This depends un the processing time the CS requires for performing various tasks such as user discovery.

Malware Detection: The security of the relationship will highly depend upon any malwares, errors, software vulnerabilities and undesirable software installations while running the JS client on the browser.

C. User-IdP Trust

The trust context in User-IdP relationship is defined as user's trust of an IdP's confidentiality, reliability and security for providing authentication services. Based on the additional trust requirements for IdP [7], we present set of essential behaviors that should be considered while evaluating the experience parameter for IdP.

Identity Encryption: In WebRTC standard, the assertions are exchanged between the communicating parties via the CS. This allows CS to extract user identity information and track user activities [5]. In order to have identity confidentiality from CS, the IdP must provide encrypted identity assertions.

Audience Protection: During P2P authentication process of WebRTC, the IdP is unable to verify the party receiving the identity assertion. This allows any unauthorized party capturing the assertion to impersonate. Authentication protocols such as OIDC may be used which has the audience protection feature to verify that the authorized party is accessing the identity assertion [27].

IdP Proxy Load Time: This indicates the time in seconds required to receive all elements from the IdP web server while loading the IdP Proxy. Delay in loading IdP proxy will lag the authentication procedure required before establishing the connection.

Information Control: This IdP feature allows a user to select the information presented in the identity assertion generated by the IdP. User can achieve confidentiality and enhance privacy by controlling the amount of information shared to their communicating participants in the identity assertions.

Authentication Delay: This delay specifies the time required for an IdP to authenticate the user and generate identity assertion. The user requires to attach the identity assertion in order to initiate a call request.

Malware Detection: Detection of any malwares, errors, software vulnerabilities and undesirable software installations while running the IdP proxy on the browser.

TABLE II
ALICE TRUST POLICY DEFINING WEIGHT SCHEME VECTOR

Truster	Trustee	S^E	S^R	S^I
Alice	User	0.1	0.4	0.5
Alice	CS	0.7	0.3	0
Alice	IdP	0	1	0

VI. USER SCENARIO

Br2Br manages the security in WebRTC calling services by evaluating trust for web browsers. The user scenario in Figure 6 illustrates how Br2Br framework helps in enhancing browser's security and user privacy. Bob and Charlie authenticate themselves to a particular IdP in order to initiate a call request to Alice via "Web2Call" calling service. Br2Br will allow Alice's browser to evaluate the amount of trust that can be invested in Bob and Charlie before accepting their call requests. The trust triples for experience, reputation and identification parameters for Bob and Charlie are presented in the 3D plot of Figure 7.

Let's suppose Alice's browser blocks call requests from users having distrust value higher than 0.5. Using Alice's trust scheme vector in Table II, the final trust vector evaluated for Bob and Charlie are $(0.22, 0.56, 0.22)$ and $(0.2, 0.21, 0.59)$ respectively. Both vectors have almost same trust values however, the browser blocks Bob call request whereas allows Charlie call request. This is due to the fact that the uncertainty factor for Charlie makes the distrust value lower than 0.5.

For the same user scenario, Figure 8 speculates the dynamic behavior of the CS "web2call". Experience and reputation parameters are used to compute trust as per the trust policy in Table II. Alice's browser by default terminates connection with any CS having trust levels below a particular threshold. At time $t = t_0$, Alice's browser detects mixed content and several attacks from the "web2call". This type of behavior decreases the experience parameter which leads the trust value to fall

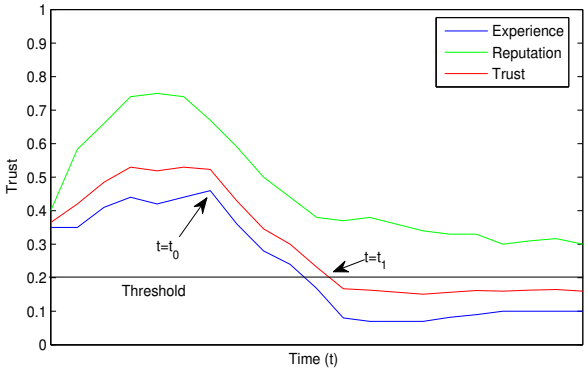

Fig. 8. Dynamic Trust Evaluation for "Web2Call"

below the threshold at time $t = t_1$. Therefore at t_1 browser will disconnect the services of "web2call" and display it to be unsafe for communication.

VII. Conclusion

We have presented a new framework for defining trust in WebRTC calling services. Our model formalizes the notion of trust, distrust and mistrust and presents three trust vector representing User-CS, User-IdP and User-User relationship in WebRTC. The framework uses three parameters namely experience, recommendation and identification to evaluate each trust vector. We propose expressions for each parameter to formalize trust in WebRTC. In our model the dependence of trust on time, context and trust policy is taken into account.

To the best of our knowledge, this model is the first where (1) formal definition of parameters relevant to WebRTC are proposed, (2) identification parameter is derived to measure the strength in user authentication process, (3) the trust context for WebRTC is described. Moreover, different attributes for trust evaluation are defined and discussed based on the security and trust requirements of WebRTC. As our future work we intend to extend this framework for WebRTC inter-operable communication services and propose appropriate P2P trust models for user-user relationship.

Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 645342, project reTHINK.

References

[1] A. Bergkvist, D. C. Burnett, C. Jennings, A. Narayanan, and B. Aboba, "WebRTC 1.0: Real-time Communication Between Browsers," W3C Working Draft, Tech. Rep., May 2016.

[2] C. Holmberg, S. Hakansson, and G. Eriksson, "Web real-time communication use cases and requirements," Internet-Draft, Tech. Rep., March 2015.

[3] V. Beltran and E. Bertin, "Unified communications as a service and webrtc: An identity-centric perspective," *Computer Communications*, vol. 68, pp. 73 – 82, 2015.

[4] A. Johnston and D. Burnett, *WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web*. Digital Codex LLC, 2012.

[5] V. Beltran, E. Bertin, and N. Crespi, "User identity for webrtc services: A matter of trust," *IEEE Internet Computing*, vol. 18, no. 6, pp. 18–25, Nov 2014.

[6] W. De Groef, D. Subramanian, M. Johns, F. Piessens, and L. Desmet, "Ensuring endpoint authenticity in webrtc peer-to-peer communication," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016, pp. 2103–2110.

[7] V. Beltran, E. Bertin, and S. Cazeaux, "Additional Use-cases and Requirements for WebRTC Identity Architecture," Internet-Draft, Tech. Rep., March 2015.

[8] R. Zhou and K. Hwang, "Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–473, April 2007.

[9] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing p2p reputation systems," *Comput. Netw.*, vol. 50, no. 4, pp. 472–484, Mar. 2006.

[10] Z. Li and J. Bi, "An adaptive trusted request and authorization model for mobile peer-to-peer networks," in *High Performance Computing and Communications(HPCC) 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, Nov 2013, pp. 1274–1280.

[11] L. Li, W. Chou, Z. Qiu, and T. Cai, "Who is calling which page on the web?" *IEEE Internet Computing*, vol. 18, no. 6, pp. 26–33, Nov 2014.

[12] L. Lpez-Fernndez, M. Gallego, B. Garca, D. Fernndez-Lpez, and F. J. Lpez, "Authentication, authorization, and accounting in webrtc paas infrastructures: The case of kurento," *IEEE Internet Computing*, vol. 18, no. 6, pp. 34–40, Nov 2014.

[13] R. L. Barnes and M. Thomson, "Browser-to-browser security assurances for webrtc," *IEEE Internet Computing*, vol. 18, no. 6, pp. 11–17, Nov 2014.

[14] N. Bruce, Y. S. Lee, S. G. Lee, and H. J. Lee, "A privacy preserving security protocol-based application for wireless communication system," in *High Performance Computing and Communications (HPCC) 2013, IEEE International Conference on Embedded and Ubiquitous Computing*, Aug 2015, pp. 1651–1656.

[15] A. Jsang, "A subjective metric of authentication," in *Proceedings of ESORICS'98, Louvain-la-Neuve*. Springer, 1998, pp. 329–344.

[16] E. Bertin, S. Cubaud, S. Tuffin, N. Crespi, and V. Beltran, "Webrtc, the day after: What's next for conversational services?" in *Intelligence in Next Generation Networks (ICIN), 2013 17th International Conference on*, Oct 2013, pp. 46–52.

[17] I. T. Javed, R. Copeland, N. Crespi, and O. al., "Global identity and reachability framework for interoperable p2p communication services," in *19th conference on Innovations in Clouds, Internet and Networks (ICIN 2016)*, March 2016.

[18] 3GPP, "Web Real-Time Communications (WebRTC) access to the IP Multimedia (IM) Core Network (CN) subsystem (IMS);," 3rd Generation Partnership Project, TR 24.371. [Online]. Available: http://www.3gpp.org/dynareport/24371.htm

[19] V. Beltran, "Characterization of web single sign-on protocols," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 24–30, July 2016.

[20] E. Rescorla, "WebRTC Security Architecture," IETF Internet Draft, Standards Track, June 2016.

[21] C. Jennings, T. Hardie, and M. Westerlund, "Real-time communications for the web," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 20–26, April 2013.

[22] I. T. Javed, K. Toumi, and N. Crespi, "Browser-to-browser authentication and trust relationships for webrtc," in *The Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies(UBICOMM)*, Oct 2016.

[23] K. Toumi, C. Andrés, A. Cavalli, and M. E. Maarabani, "A vector based model approach for defining trust in multi-organization environments," in *7th Int. Conf. on Risks and Security of Internet and Systems, CRISIS'12*. IEEE Computer Society Press, 2012, p. in press.

[24] I. Ray and S. Chakraborty, "A vector model of trust for developing trustworthy systems," in *Computer Security–ESORICS 2004*. Springer, 2004, pp. 260–275.

[25] W. E. Burr, D. F. Dodson, and W. T. Polk, *Electronic authentication guideline*. Citeseer, 2004.

[26] E. . Rescorla, "Security Considerations for WebRTC," IETF Internet Draft, Standards Track, July 2013.

[27] J. Bradley, B. de Medeir, and C. Mortimore, "Openid connect core 1.0," *The OpenID Foundation*.