PROVER.

WHITE PAPER

CÔNG NGHỆ BẢO MẬT XÁC MINH BẢN QUYỀN VIDEO

Nội dung

1. Giới thiệu

Dịch vụ PROVER ra đời nhằm thực hiện việc xác minh quyền sở hữu bằng nội dung video một cách trực quan và độc lập. PROVER là một dịch vụ trực tuyến sử dụng công nghệ Blockchain, cho phép người sử dụng có thể xác nhận quyền sở hữu Video bằng cách nhận diện các chuyển động và sự kiện xảy ra trong đoạn Video đó. Ngoài ra, PROVER còn có thể được sử dụng khi không có kết nối mạng, hỗ trợ nhúng mã, chia sẻ trên các công cụ, nền tảng khác cũng như được đưa vào ứng dụng trong các lĩnh vực liên quan dựa trên các tính năng riêng biệt của nó.

2. Tổng Quan

Rõ ràng là cuộc cách mạng công nghệ chuyển sang sử dụng điện thoại thông minh được khởi xướng bởi sự xuất hiện của iPhone và máy tính bảng cách đây 10 năm đã ảnh hưởng đáng kể đến chúng ta trong quan điểm sử dụng điện thoại để liên lạc và tương tác. Một cách ngẫu nhiên, bất cứ đều có thể trở thành một người sáng tạo cũng như xây dựng các kịch bản cho các đoạn phim ngắn mà kết quả hiển nhiên là số lượng này có xu hướng gia tăng ngày một nhiều hơn. Nhiều lĩnh vực liên quan trong đó có kinh tế chịu ảnh hưởng nhiều nhất hệ quả của quá trình công nghệ kỹ thuật số này. Hình ảnh và các đoạn phim ngắn giờ đây không chỉ được sử dụng cho một mục đích là giải trí hay giáo dục nữa mà chúng còn được sử dụng cho các mục đích khác ví dụ như trong kinh tế và các lĩnh vực cần đến pháp lý như tài chính, bảo hiểm, tư pháp, y tế và các dịch vụ khác. Chính vì vậy, việc xây dựng một hệ thống xác nhận bảo mật bản quyền nội dung Video một cách độc lập, trực quan chống lại việc chỉnh sửa, sao chép và làm giả là điều vô cùng cần thiết.

Giải pháp cho vấn đề nêu trên có thể là bàn đạp cho sự phát triển các ngành kinh tế ứng dụng công nghệ kỹ thuật số. Nơi mà công nghệ có thể được sử dụng trong hàng nghìn dự án từ hàng chục khu vực khác nhau và đem lại trải nghiệm tích cực cho hàng trăm triệu người.

3. Hiện trạng

Tính xác thực của các đoạn phim ghi lại các sự kiện, và sự thật mang tính giá trị về kinh tế và pháp luật luôn là một nghi vấn lớn bởi các dữ liệu video có thể bị sửa đổi cho các mục đích sai trái bằng cách sử dụng các camera ảo (camera giả lập). Những thông tin về ngày tháng tạo ra video cũng có thể được làm giả bởi con người.

PROVER ra đời với nhiệm vụ là bảo đảm tính xác thực của việc ghi lại cụ thể thời gian hình thành một đoạn video.

4. Giải pháp đề xuất

4.1. Giải pháp bảo vệ bản quyền nội dung trực tuyến

Hầu hết các công cụ giúp xác minh quyền sở hữu, bản quyền của một nội dung (video) ngày nay chỉ có thể phê duyệt ở mức độ cơ bản trong lưu trữ và xác thực các thông tin dữ liệu về đoạn video. Một ví dụ điển hình của giải pháp này là sử dụng Xác nhận danh tính nội dung của Google (Google content ID), tính năng này hoạt động nhờ vào việc xác nhận quyền sở hữu bản gốc đoạn video bằng cách đối chiếu thời gian tải đoạn video đó lên Youtube (tức là hệ thống này sẽ so sánh thời gian đoạn video này được tải lên với các video được đăng lên trước đó và mặc định bản quyền cho những video được đăng tải lên trước). Phương thức này hoạt động dựa trên nguyên tắc giả định quyền tác giả - một người được xem như là tác giả của tác phẩm cho đến khi tác giả thật bác bỏ điều này).

Mặt trái của các giải pháp tương tự như thế này là người ta không thể lưu lại thời gian thực, bản gốc và sự nguyên vẹn của đoạn phim. Hơn nữa, giải pháp này bị hạn chế bởi chỉ có thể ứng dụng trên Youtube và chịu sự tác động bởi các Quản trị viên. Điều này đồng nghĩa với việc có sự tác động trực tiếp từ yếu tố con người một cách chủ quan hoặc cố tình trong việc xét duyệt bản quyền. Với một số người muốn hiểu sự thật với thái độ khó chịu, đoạn đối thoại phản hồi sẽ rất ngắn gọn - "Người sở hữu nội dung sẽ bị tước quyền sử dụng Content ID và mất trạng thái đối tác với Youtube nếu như họ khiếu nại một cách liên tục và vô lý".

Các tác giả cố gắng bảo vệ nội dung của họ bằng cách thêm các hình mờ hoặc biểu tượng trên hình chiếu video nhưng thực tế là chúng chỉ có thể giúp họ trong các cuộc tranh cãi về mặt bản quyền sau đó, chứ không ngăn chặn được việc bị giả mạo. Điều này cũng có thể được hiểu là trong các vấn đề liên quan tới pháp lý và tài chính, giải pháp này hoàn toàn không thể sử dụng được.

4.2. Giải pháp bảo vệ bản quyền thông qua nền tảng Blockchain

Tính đến thời điểm hiện tại, cộng đồng Blockchain đã cho ra đời nhiều dịch vụ công chứng điện tử khác nhau như "Proof of Existence" là bằng chứng xác nhận cho sự hiện diện, hiển thị hay "Proof of Ownership" để bảo vệ chủ quyền đối với mọi loại tài liệu và dữ liệu kỹ thuật số:

Block Notary – một dịch vụ giúp người sử dụng tạo ra "Proof of Existence" – là chứng minh sự tồn tại của các dữ liệu có định dạng như: tệp ảnh, thư mục hoặc bất kỳ công cụ chuyển tải nội dung nào sử dụng hệ thống TestNet 3 hoặc Bitcoin. Hệ thống giao diện tương tác với người sử dụng điện thoại dành cho hệ điều hành iOS được ghi lại dưới dạng một tài liệu hàm băm trong chuỗi khối blockchain.

Emercoin DPO Antifake - Công nghệ ứng dụng nền tảng Erner cho phép người sử dụng tạo ra một hộ chiếu kĩ thuật số riêng biệt lưu trữ trong hệ thống cơ sở dữ liệu phân tán - blockchain, và thực hiện các dịch vụ quản lý quyền sổ hộ chiếu này. Emercoin DPO Antifake tập trung chủ yếu vào các phân khúc thị trường ngoại tuyến (không có mạng Internet) - ứng dụng này giúp người dùng đăng kí các thông tin cá nhân (số khung xe ô tô, IMEI - số nhận dạng thiết bị điện thoại) trong hệ thống nhằm bảo vệ hàng hoá không bị gian lận.

Stampery - sử dụng công nghệ Blockchain có khả năng lưu trữ, xác thực mọi e-mail và thư mục dữ liệu.

Hệ thống này làm tối giản quá trình xác thực các chữ cái bằng cách gửi chúng tới một địa chỉ email được tạo riêng cho mỗi khách hàng. Các công ty luật sử dụng Stampery để tiết kiệm chi phí trong việc gửi thư xác nhận đến từng khách hàng riêng biệt.

https://www.ascribe.io - Dịch vụ đăng kí bản quyền kèm với quản lý và phân phối các nội dung kỹ thuật số. Nó được ứng dụng trong ngành Mỹ thuật bằng cách cung cấp các mã kĩ thuật số cho các tác phẩm nghệ thuật khi đem ra trưng bày, đấu giá hoặc bày bán trong một thị trường có độ bảo mật cao.

https://letsnotar.me - Một ứng dụng tự động lưu giữ mọi thông tin người dùng đăng tải lên chuỗi khối (blockchain) bằng một mã hàm băm. Ứng dụng này có thể sử dụng trên điện thoại, được truy cập vào máy ảnh, chụp ảnh và tạo nên video và lưu trữ chúng dưới dạng một mã hàm băm. Tuy nhiên, ứng dụng này lại không đảm bảo được liệu video ghi lại được tạo ra từ một máy ảnh thật hay máy ảnh ảo - và vì vậy, nó không thể chống lại được việc làm giả video.

Tất cả các dịch vụ trên đề có một điểm chung: chúng đều có thể bảo đảm cho mã hàm băm của các dữ liệu có sẵn hoặc được cho là được tạo bằng các thiết bị ghi hình nhưng chúng không thể bảo đảm được tính thuần nhất, nguyên vẹn và bản quyền của các đoạn video. Chúng không thể chống làm giả hoặc sửa đổi chủ quan từ tác động bên ngoài bởi công nghệ mà các ứng dụng này sử dụng không thích hợp để xác minh được nguồn gốc của các đoạn phim được tạo ra. Công nghệ của PROVER có khả năng chắc chắn nội dung video được tạo ra từ một thiết bị ghi hình cụ thể, với mốc thời gian rõ ràng và bảo đảm video nguyên bản không có dấu hiệu bị chỉ nh sửa hoặc gian lận.

PROVER có khả năng đảm bảo 100% tính xác thực bản quyền khi tạo ra một đoạn video. PROVER cũng có thể trở thành một chức năng bổ sung giúp nâng cao lòng tin của người sử dụng trong các lĩnh vực cần thiết, một trong các lĩnh vực ứng dụng được công nghệ này đó là cung cấp dịch vụ công chứng bản quyền của những video được phát hành.

Cho đến nay, chúng tôi vẫn chưa có bất kỳ thông tin nào về các dự án khác có liên quan đến lĩnh vực xác thực bản quyền video tương tự như thế được nghiên cứu và phát triển. Điều này hoàn toàn không có gì đáng kinh ngạc bởi dịch vụ này đòi hỏi sự kết hợp giữa công nghệ blockchain và kĩ thuật xây dựng, nâng cấp các mô-đun xử lý video. Điểm mạnh của chúng tôi nằm ở nhân sự, bởi đội ngũ phát triển PROVER là sự kết hợp tất cả các kinh nghiệm, kỹ thuật cần thiết cho lĩnh vực phức tạp này.

5. PROVER là gì?

Dịch vụ PROVER bao gồm một số các công cụ kết hợp sau:

- Một ứng dụng điện thoại được cài trên điện thoại thông minh và kết nối với máy ảnh điện thoại khi máy ảnh được bật hoặc khởi động để sử dụng.

- Một bộ thuật toán và các tính năng tiện ích tích hợp công nghệ PROVER dành cho các ứng dụng, dịch vụ của bên thứ 3.

- Hợp đồng thông minh PROOF (chỉ dành cho các công cụ, ứng dụng hoạt động dựa trên nền tảng của Ethereum).

Vấn đề cốt lõi cần được giải quyết của dự án chính là việc xác minh được nguồn gốc của video do người sử dụng chụp. Trong quá trình xác minh, hệ thống sẽ xác nhận:

1. Đoạn Video được hình thành thông qua một thiết bị ghi hình thật chứ không phải được dựng lên bởi một ca-me-ra ảo.

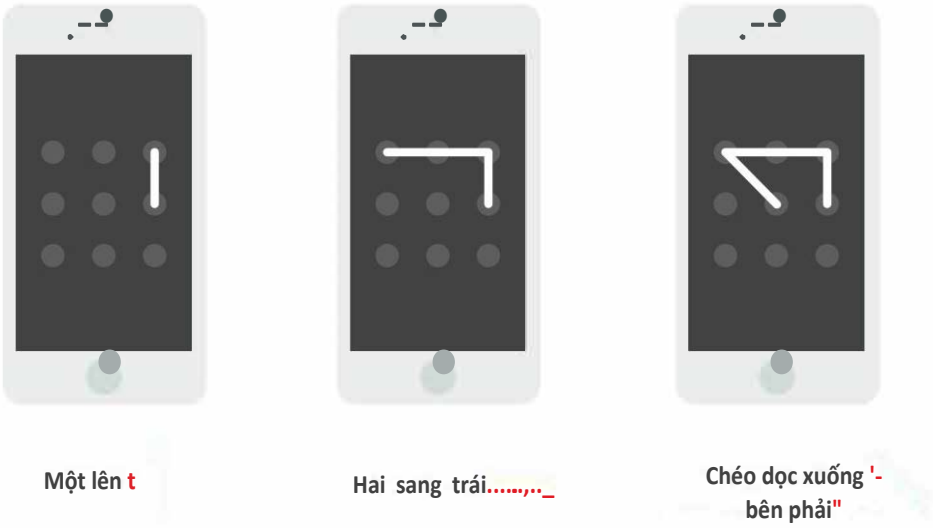
2. Đoạn Video được hoàn thành mà không có dấu hiệu bị sửa chữa, cắt, dán & chèn thêm.

3. Bản thu hình được mở trong một khoảng thời gian xác định.

Mọi thông tin mã nguồn của dự án sẽ được lưu trữ tại trang trực tuyến:
https://github.com/isvirin/prover.

5.1. Xác thực bằng bộ mã SWYPE

Thuật toán tự động phát hiện mã SWYPE trong các khung hình video sẽ là cơ sở xác minh video cơ bản đầu tiên. Thay vì sử dụng thuật toán SWYPE truyền thống: dùng ngón tay nối liền các điểm trên màn hình một cách liên tục. Trong công nghệ PROVER, đoạn mã SWYPE sẽ được ghi nhận dữ liệu bằng cách di chuyển điện thoại ở chế độ ghi hình. Trong dự án, các mô-đun phân tích video sẽ được phát triển đầu tiên. Các mô-đun này giúp phân tích và phát hiện dòng chuyển động của máy ảnh điện thoại, nhờ vào đó, hệ thống có thể xây dựng được một mã SWYPE ảo bắt buộc phải nối thành 3 đường thẳng liên tục, mỗi đường phải có 3 điểm nút.



Nói cách khác, khi sử dụng máy ảnh trên điện thoại, người sử dụng sẽ nhìn thấy nút bấm với dòng chữ "Vui lòng nhập mã SWYPE" trên màn hình, khi chạm vào đó, người dùng sẽ nhìn thấy một khung vuông gồm 9 điểm nối: 3 điểm nối mỗi hàng.

Một mã SWYPE tự động được tạo ra cho đầu vào trong chuỗi khối (blockchain). Điểm đầu tiên chính là nơi bắt đầu dòng chuyển động. Để nhập mã SWYPE, người dùng cần di chuyển điện thoại trong không gian để dòng mã SWYPE di chuyển theo đường dẫn trên màn hình. Sau khi mã swype được nhập, khung nhập mã biết mất và xuất hiện thông báo rằng đã nhập mã thành công.

Sử dụng thuật toán tự động trong nhận dạng mã Swype giúp người sử dụng có thể thoải mái quay video sau khi nhập mã. Sau đó, nếu đoạn video cần kiểm tra xác minh bản quyền, mã SWYPE sẽ được PROVER chấp nhận.

5.2. Xác thực thông qua bộ cảm biến

Một hệ thống siêu dữ liệu (dữ liệu từ tất cả các nguồn cảm biến có trong thiết bị điện thoại cầm tay, máy gia tốc, cảm biến con quay hồi chuyển, máy đo từ, tọa độ GPS,... sẽ hoạt động với công suất tối đa) sẽ được ghi lại cùng lúc với đoạn video để tránh bị giả mạo. Sau đó, chúng tôi sẽ phát triển một thuật toán cho phép lấy lại được toàn bộ dữ liệu này một cách chi tiết và chính xác nhất về các chuyển động của thiết bị điện thoại trong tay người sử dụng với những chuyển động của video.

Chúng tôi sẽ sử dụng mô hình toán không gian ba chiều dựa trên một gia tốc cảm biến 3 trục và từ kế 3 trục với điều kiện sự di chuyển có gia tốc không đáng kể.

Để xác đị nh vị trí của vật trong không gian, chúng tôi xin được giới thiệu hệ trục toạ độ $OXYZ$, vì vậy ta có trục OZ, trùng với hướng của trọng lực $\vec{g}$, và trục OY, trùng với mặt phẳng nghiêng chứa véc tơ $\vec{h}$ - lực trọng trường trái đất.

Để mô tả vị trí cảm biến trong hệ tọa độ địa lý (GCS), chúng tôi xin được giới thiệu hệ tọa độ địa phương (LCS), các trục của hệ LCS sẽ trùng với các trục của gia tốc cảm biến và từ trường. Vị trí cảm biến của LCS trong GSC được mô tả bởi 4 véc-tơ sau:

$\overrightarrow{r_{LCS}}$ – véc-tơ chuyển vị của hệ trục LCS so với hệ trục GCS gốc;

$\overrightarrow{\imath_{LCS}}$, $\overrightarrow{\jmath_{LCS}}$, $\overrightarrow{k_{LCS}}$ – véc-tơ có hướng của hệ trực chuẩn LCS được thể hiện dưới dạng các véc-tơ có hướng của hệ trực chuẩn GCS.

Thông tin mô tả đã cho thấy một cách rõ ràng và đầy đủ sự định hướng của hệ trục LCS so với hệ trục gốc GCS. Vấn đề xác định hướng góc được quy về việc đi tìm tọa độ của các véc tơ $\overrightarrow{\imath_{LCS}}$, $\overrightarrow{\jmath_{LCS}}$, $\overrightarrow{k_{LCS}}$ trong hệ trục GCS.

Trong thực tế, trường trọng ổn định hơn từ trường, ta bắt đầu với việc lấy cơ sở của một gia tốc cảm biến. Việc đọc gia tốc cảm biến chính là đọc tọa độ của các gia tốc rơi tự do, phân tích theo các trục của LCS thì:

$$\overrightarrow{a_{LCS}} = \{a_x, a_y, a_z\}$$

Véc-tơ chuẩn $\overrightarrow{a_{LCS}}$ bằng không ngược lại véc-tơ $\overrightarrow{k_{GCS}}$, i.e., được xác định bởi trục OZ GSK, được biểu diễn dưới dạng các véc-tơ có hướng của hệ trục LCS:

$$\overrightarrow{k_{\text{rск}}} = \frac{\overrightarrow{a_{LCS}}}{|\overrightarrow{a_{LCS}}|} = \{k_x, k_y, k_z\}$$

Các chỉ số cảm biến từ trường là tọa độ của véc-tơ từ trường, phân tích theo trục LCS thì:

$$\overrightarrow{m_{LCS}} = \{m_x, m_y, m_Z\}$$

Thông thường, véc-tơ $\overrightarrow{m_{LCS}}$ sẽ không song song với véc-tơ $\overrightarrow{k_{GCS}}$, vì vậy nó có dạng như sau:

$$\overrightarrow{n_{LCS}} = \overrightarrow{m_{LCS}} - (\overrightarrow{m_{LCS}} \cdot \overrightarrow{k_{GCS}}) \cdot \overrightarrow{k_{GCS}}$$

Véc-tơ chuẩn $\overrightarrow{n_{LCS}}$ bằng không ngược lại với véc-tơ $\overrightarrow{J_{GCS}}$, i.e. được xác định bởi trục *OY SK được biểu diễn dưới dạng các véc-tơ có hướng của hệ trục LCS:*

$$\overrightarrow{J_{\text{rск}}} = \frac{\overrightarrow{n_{LCS}}}{|\overrightarrow{n_{LCS}}|} = \{j_x, j_y, j_z$$

Trục véc-tơ OX GCS được biểu diễn dưới dạng véc-tơ có hướng của trục LCS, được thể hiện bằng tích của hai véc-tơ:

$$\overrightarrow{\imath_{GCS}} = \overrightarrow{J_{GCS}} \times \overrightarrow{k_{GCS}} = \{i_x, i_y, i_z\}$$

Ta có được các ma trận hàng của các véc-tơ sau: $\overrightarrow{\imath_{GCS}}$, $\overrightarrow{J_{GCS}}$, $\overrightarrow{k_{GCS}}$ tiếp tục chuyển vế và khai triển chúng thành dạng véc-tơ (vẫn là hàng ngang):

$$|i_x\ i_y\ i_z\ j_x\ j_y\ j_z\ k_x\ k_y\ k_z\ |^T = |i_x\ j_x\ k_x\ i_y\ j_y\ k_y\ i_z\ j_z\ k_z\ |$$

Từ đó, ta có các véc-tơ:

$$\overrightarrow{\imath_{LCS}} = \{ix, jx, kx\}$$

$$\overrightarrow{J_{LCS}} = \{iy, jy, ky\}$$

$$\overrightarrow{k_{LCS}} = \{i_z, j_z, k_z\}$$

Xác định trục của LCS trên hệ trục GCS, nói cách khác, xác định hướng của LCS trên GCS.

Ta biểu thị véc-tơ $\overrightarrow{a_{LCS}}$ trên trục GCS:

$$\vec{a} = a_x \cdot \overrightarrow{\iota_{LCS}} + a_y \cdot \overrightarrow{J_{LCS}} + a_z \cdot \overrightarrow{k_{LCS}} = \{a'_x, a'_y, a'_z\}$$

Giá trị của véc-tơ $\vec{a}$ cho ta thấy lượng tử của ADC trong bộ cảm biến gia tốc, để tính toán thuận tiện hơn, ta chuyển chúng sang Hệ thống lượng tử Quốc tế (ISQ), công thức gia tốc tức thời lúc này là:

$$\overrightarrow{a_{phys}} = \frac{range}{N} \cdot \vec{a},$$

"Range" được hiểu là biên độ dao động của bộ chuyển đổi ADC, N là số bị chia. Để tính trọng lực, trong trường hợp này, ta đưa về véc-tơ thành phần song song với trục tung bằng cách tính giá trị của gia tốc trọng lực:

$$\overrightarrow{a_g} = \overrightarrow{a_{phys}} - \{0,0,-g\}$$

Chuyển giá trị gia tốc sang vận tốc ta được:

$$\overrightarrow{v_g} = \int \overrightarrow{a_g} dt + \overrightarrow{v_0} \approx \sum \overrightarrow{a_g} \Delta T + \overrightarrow{v_0}$$

Và chuyển từ vận tốc sang vị trí tọa độ:

$$\overrightarrow{r_g} = \int \overrightarrow{v_g} dt + \overrightarrow{r_0} \approx \sum \overrightarrow{r_g} \Delta T + \overrightarrow{r_0}$$

Như vậy, thông qua mô hình toán học trên, ta có thể xác đị nh hướng của bộ cảm biến tương đối so với hệ thống GCS bằng cách sử dụng các đặc tính vật lý để tính toán các giá trị cảm biến, và xác đị nh bằng các tích hợp cảm biến các chuyển động tuyến tính trong GCS. Công việc còn lại là thu nhập kết quả đo lường được và khôi phục lại số liệu chuyển động của người dùng thông qua cách tính trên.

6. Cách thức hoạt động

6.1. Thuật toán chung

Người dùng cài phần mềm PROVER vào điện thoại, cho phép ứng dụng được tương tác với máy ảnh cho phép phần mềm được khởi động tự động khi camera bật. Khi bật máy ảnh, người sử dụng truy cập vào hệ thống Blockchain bằng mạng Internet để nhận mã SWYPE và thực hiện các bước xác minh mã trước khi thiết lập dữ liệu và ghi lại đoạn phim. Nhập mã SWYPE khi bật chết độ ghi hình bằng cách chuyển động điện thoại một cách ngẫu nhiên theo 3x3 nút hiển thị trên màn hình.

Mỗi lần sử dụng chức năng ghi hình trên điện thoại, người sử dụng cần phải nhập mã SWYPE. Người dùng có thể nhận biết các điểm ảo 3x3 trên màn hình khi thông báo nhập mã SWYPE hiện lên. Để đảm bảo hơn, chúng tôi phát triển hệ thống đồng bộ hóa các chuyển động thông qua bộ cảm biến của điện thoại thông minh (gia tốc kế và từ trường kế).

Sau khi video được hoàn thành, hàm băm dữ liệu của video được hoàn tất và lưu giữ trong hệ thống blockchain.

Video được ghi lại được lưu giữ bởi người sử dụng hoặc các thiết bị cá nhân hoặc thông qua phương pháp lưu trữ điện toán đám mây và được sử dụng để chứng minh bản quyền trong trường hợp cần thiết.

Vì vậy mà các tập dữ liệu được lưu trữ trong chuỗi blockchain có những thông tin sau:

- Ngày, giờ khi người sử dụng nhập mã swype;
- Mã swype do người dùng tạo ra được lưu lại;
- Một hàm băm lưu trữ dữ liệu của tệp video trên hệ thống blockchain (đoạn video vẫn được người dùng lưu lại riêng)
- Thời gian tạo ra mã hàm băm.

Để chứng minh quyền sở hữu một đoạn video, chúng ta chỉ cần tính toán mã hàm băm đã được lưu lại trên hệ thống blockchain để nhận được dữ liệu; nếu không có mã hàm băm, ta nhận được thông báo hàm băm chưa được tải lên. Trên cơ sở thông tin nhận được, thông tin cho thấy mã hàm băm của dữ liệu được tạo ra trong khoảng thời gian sau khi người dùng hoàn thành đoạn mã swype của riêng họ và trước khi nó được tải lên hệ thống lưu trữ blockchain.

Tập video được người dùng tải lên sẽ được kiểm tra liên tục để loại trừ tình huống bị thay đổi và xác nhận mã swype đi kèm. Sau khi hoàn thành giai đoạn đầu tiên của dự án PROVER, mã swype sẽ được nhận diện bằng thuật toán tự động. Tiếp đó, thuật toán tự động sẽ kiểm tra tính liên tục của đoạn video. Nếu tồn tại dữ liệu của đoạn video trong khối blockchain có cùng mã swype, hệ thống có thể kết luận rằng bản ghi video được tạo ra bởi người dùng sau khi đăng kí mã swype. Đây là thời điểm mà mã swype đã được lưu trữ và không thể sửa chữa bởi người dùng.

Sau khi hoàn thành giai đoạn thứ 3, PROVER được phát triển một thuật toán giúp hỗ trợ trong việc tái thiết lại các chuyển động của điện thoại người dùng cầm trên tay khi ghi hình và với quỹ đạo chuyển động được thiết bị ghi lại. Nếu hai xu hướng chuyển động trên trùng khớp nhau thì PROVER sẽ đưa ra kết luận cuối cùng rằng video được tạo ra hoàn toàn từ máy ảnh thật chứ không phải từ một camera ảo hoặc làm giả.

6.2. Sử dụng nền tảng Ethereum

PROVER sử dụng nền tảng Ethereum để tạo nên hợp đồng thông minh PROOF. Thiết bị điện thoại truy cập hệ thống nhờ vào mạng In-ter-net đến hợp đồng thôn minh PROOF để nhận mã SWYPE.

Hợp đồng thông minh PROOF lưu lại thông tin mỗi lần phát hành mã SWYPE và ghi lại cụ thể thời gian phát hành. Sau khi video được hoàn thành và hàm băm được nhận biết, chúng được gửi đi và lưu giữ trên hợp đồng thông minh PROOF.

Kèm theo đó là các thông tin:

1. Thời gian phát hành mã SWYPE;
2. Mã SWYPE được phát hành;
3. Một hàm băm lưu trữ dữ liệu của tệp video trên hệ thống blockchain (đoạn video vẫn được người dùng lưu lại riêng);
4. Thời gian tạo ra mã hàm băm;

Để xác thực bản quyền của video, hệ thống tính toán mã hàm băm và gửi cho hợp đồng thông minh PROOF. Đáp lại, hợp đồng thông minh sẽ thông báo tình trạng sử dụng của mã SWYPE (đã được sử dụng hay chưa) và cho phép truy cập mã SWYPE nếu không bị trùng mã trong hệ thống lưu giữ. Dựa trên thông tin nhận được, mã hàm băm của dữ liệu được tạo ra trong khoảng thời gian sau khi người dùng hoàn thành đoạn mã swype của riêng họ và trước khi nó được tải lên hệ thống lưu trữ blockchain

6.3. Sử dụng nền tảng Emercoin

Nền tảng Emercoin giả định rằng mã Swype được tạo thành dựa trên mã hàm băm cuối cùng từ các khối trong chuỗi khối, địa chỉ khách hàng trên hệ thống blockchain sẽ kèm theo các thông tin như số IMEI của điện thoại khách hàng. Đoạn mã mô phỏng để tạo mã SWYPE có dạng như sau:

```
bytes32 mixedString = mix(userAddress, IMEI, blockHash);
bytes32 temp  = sha3(mixedString);
swype = uint16(uint256(temp)  % 65536);
```

Hàm băm của video được hình thành sau khi người dùng quay xong video, chúng sẽ được lưu trữ trong hệ thống blockchain của Emercoin cùng với số khối có mã SWYPE, cũng như số IMEI của thiết bị điện thoại đã sử dụng để tạo video, sự trao đổi giữa các hệ thống cần phải được thực hiện trên cùng máy điện thoại đã dùng để tính toán mã SWYPE.

Để xác minh bản quyền của đoạn video, hệ thống sẽ tính toán mã hàm băm lưu trữ thông tin trên hệ thống chuỗi khối Emercoin. Trên cơ sở thông tin được lưu giữ, hệ thống có thể kết luận rằng mã hàm băm này được tạo ra cùng với thời gian được ghi lại trên các khối trong chuỗi khối ghi lại hàm băm của đoạn mã SWYPE và trước khi mã hàm băm của dữ liệu video được lưu giữ trên hệ thống blockchain.

7. Các lĩnh vực ứng dụng

Mặc dù công nghệ PROVER có thể được sử dụng một cách riêng biệt, tuy nhiên, chúng lại sở hữu một số giá trị đặc trưng có khả năng hỗ trợ và xây dựng nền tảng cho rất nhiều các ứng dụng, dịch vụ thuộc nhiều ngành nghề khác nhau.

7.1. Bảo hiểm ô tô

Ta có thể thấy được việc tạo ra một ứng dụng trong bảo hiểm xe hơi là hoàn toàn có khả năng, ví dụ như bảo hiểm cho chuyến đi ngắn ngày (có thể là một ngày chẳng hạn). Khách hàng ghi đoạn phim về tình trạng cụ thể của xe khi kí hợp đồng bảo hiểm, đoạn phim này được chứng nhận bởi PROVER và trong trường hợp xảy ra tai nạn hoặc tranh chấp, nó sẽ trở thành bằng chứng trước công ty bảo hiểm.
Các thành phần có thể được tích hợp cho ứng dụng của công ty bảo hiểm:

- Nền tảng hệ thống nhận yêu cầu từ ứng dụng điện thoại khách hàng chuyển chúng tới hệ thống công ty bảo hiểm và nơi lưu trữ thông tin trong chuỗi blockchain.

- Nhờ đó, máy chủ tại công ty bảo hiểm sẽ đánh giá và cung cấp dịch vụ cho khách hàng của họ. Trong thực tế, các máy chủ công ty bảo hiểm sẽ tính toán xem lượng tiền mà khách hàng phải chi ra cho dịch vụ này. Với các dữ liệu được xác thực, các máy chủ của công ty bảo hiểm có thể truy cập vào hệ thống và chuyển trực tiếp vào hệ thống lưu giữ blockchain.

- Ứng dụng điện thoại sẽ quản lý tình trạng của Khách hàng, cho phép khách hàng kích hoạt hoặc tắt bảo hiểm, đặc biệt là trong việc giới hạn số lượng dịch vụ cung cấp. Ứng dụng này sẽ chỉ tương tác với máy chủ nền tảng - nơi lưu trữ thông tin khách hàng trên hệ thống blockchain.

Mã truy nhập tới hệ thống có thể chính là số hợp đồng bảo hiểm được khách hàng mua tại công ty bảo hiểm. Khi chức năng bảo hiểm được kích hoạt, người sử dụng phải quay lại tình trạng của xe, các góc và toàn bộ mặt xe. Video của người dùng có thể được lưu trong điện thoại, máy tính bảng, lưu trữ trực tuyến như Google và Dropbox và một mã hàm băm sẽ được gửi tới hệ thống máy chủ. Trong trường hợp bảo hiểm xảy ra, khách hàng gửi tệp video cho công ty bảo hiểm và công ty sẽ xác minh video bằng mã hàm băm và thời gian lưu giữ video trên blockchain.

Giải pháp ứng dụng này cho phép công ty bảo hiểm tránh được tình trạng khách hàng gian lận bằng cách thông đồng với nhân viên bảo hiểm hoặc một công chức thi hành luật làm giả một hợp đồng hư hại xe hoặc yêu cầu bồi thường cho một sự kiện giả được sắp đặt từ trước. Trên phạm vi toàn cầu, ứng dụng cho phép các công ty luật tiết kiệm hàng tỷ đô-la mỗi năm.

Điều quan trọng ở đây là công nghệ của chúng tôi cho phép khách hàng tin tưởng tuyệt đối vào hệ thống ghi nhận bởi mọi dữ liệu được ghi lại là hoàn toàn trùng khớp với thời gian xảy ra các sự kiện. Hơn thế nữa, dữ liệu video được ghi lại thông qua thiết bị điện tử cá nhân của chính khách hàng (điện thoại cầm tay, máy tính bảng, máy tính cá nhân, máy tính xách tay), giúp loại bỏ khả năng video được quay trước khi hợp đồng bảo hiểm được kí hoặc cố tình bị làm giả để gian lận.

PROVER sẽ trở thành một nền tảng cũng như một ứng dụng mang tính đột phá cho phép người sử dụng bật hoặc tắt hợp đồng bảo hiểm hoặc đăng kí các gói bảo hiểm khác nhau. Thêm vào đó, ứng dụng này giúp người chủ xe trong việc xác định và thiết lập các hợp đồng bảo hiểm sẵn có và đảm bảo mà không cần phải lắp đặt thêm các thiết bị theo dõi trong trường hợp xe bị mất hoặc hư hại.

Lợi thế cạnh tranh của ứng dụng này gồm có:

- Giúp người sử dụng bảo hiểm giảm tải các chi phí khi tham gia bằng cách tạo ra các tính năng quản lí gói bảo hiểm mà họ sở hữu trong suốt thời gian hợp đồng bảo hiểm diễn ra, một điểm cộng nữa là chúng kích thích nhiều người sử dụng bảo hiểm hơn bởi thông thường chi phí cao là rào cản lớn trong việc quyết đị nh sử dụng các gói bảo hiểm;
- Không cần phải cài đặt thêm các thiết bị , ứng dụng theo dõi bên trong xe;
- Việc chứng minh và xác nhận tính chân thực của bằng chứng thông qua hệ thống PROVER trở nên thuận tiện và đáng tin cậy hơn cho cả hai phía: người sử dụng và công ty bảo hiểm.

- Công ty sử dụng ứng dụng của PROVER sẽ trở nên an tâm hơn bởi mọi thông tin giao dịch của khách hàng và các hợp đồng bảo hiểm đều được ghi chép lại một cách cẩn thận và lưu trữ trong hệ thống blockchain với mức độ bảo mật cao không chỉ cho một vài khách hàng, công ty nhỏ lẻ mà là cho hàng triệu người trên toàn thế giới.

7.2. Xác thực quyền sở hữu Video bản gốc (sử dụng công nghệ kĩ thuật số)

Khi người sử dụng dùng điện thoại thông minh hoặc máy ảnh để ghi lại video, họ có thể đăng kí quyền sở hữu và được xác nhận bởi hệ thống PROVER. Điều này có thể gây thích thú cho những người sử dụng như phóng viên phải di chuyển liên tục, blogger, vận động viên, du khách, nhạc sĩ , nhà soạn nhạc hoặc bất kì ai hoạt động trong lĩ nh vực sáng tạo, nghệ thuật cần sử dụng đến mạng xã hội và các dị ch vụ tạo video.

Người sử dụng có thể dùng thử với gói miễn phí có thể mở tài khoản và đăng kí xác thực một số các video miễn phí. Sau đó ứng dụng sẽ gợi ý và cung cấp thêm các dị ch vụ để người sử dụng có thể đăng kí trả phí trong một khoản thời gian cố đị nh hoặc mua các gói sử dụng như 50, 100, 500 video.

7.3. Theo dõi bệnh nhân từ xa (Kiểm tra đơn thuốc)

Công nghệ PROVER có thể trở thành một cơ sở trong việc kiểm soát và hỗ trợ trong điều trị và theo dõi bệnh nhân từ xa, cụ thể hơn, trong việc xác nhận bệnh nhân dùng thuốc đúng với qui đị nh. Để sử dụng tính năng này, bệnh nhân có thể cài đặt ứng dụng của công ty dược hoặc phòng khám của họ có tích hợp sẵn nền tảng công nghệ PROVER. Người bệnh sẽ có một "hộp thuốc" và lị ch trình sử dụng thuốc theo đơn khám bệnh. Đến đúng giờ, một cửa sổ thông báo sẽ hiển thị , người sử dụng bật ứng dụng, nhập mã SWYPE và bắt đầu ghi lại đoạn phim ngắn trong lúc lấy thuốc và sử dụng.

Hệ thống này có thể giúp các phòng khám và công ty bảo hiểm đảm bảo được bệnh nhân 100% được điều trị đúng liều lượng và sử dụng thuốc hợp lí.

Đối với bệnh nhân, video sẽ là bằng chứng cho việc họ sử dụng thuốc đúng với đơn khám của bác sĩ và việc điều trị sẽ được công ty bảo hiểm xác nhận thanh toán.

Và cuối cùng, các công ty dược sẽ xác thực được việc bệnh nhân của họ sử dụng thuốc đúng với đơn kê của bác sĩ và tăng hiệu quả sử dụng thuốc cũng như nâng cao lợi nhuận nhờ vào sự gia tăng từ số lượng người tin tưởng và đến khám.

7.4. Chơi Game Online và làm nhiệm vụ

Công nghệ của PROVER có thể được tích hợp trong các ứng dụng chơi game giúp người sử dụng đăng kí vào các sự kiện game hay truy tìm kho báu để nhận phần thưởng và nâng cấp độ,…

7.5. Báo cáo tiến độ công việc

Công nghệ PROVER cho phép việc theo dõi và giám sát tiến độ công việc dễ dàng hơn, đặc biệt là trong các lĩ nh vực: xây dựng, lắp đặt, vệ sinh, tuần tra, giao hàng, chuyển phát nhanh, làm biển quảng cáo, v.v). Video từ PROVER sẽ đảm bảo được độ khách quan về thời gian ghi nhận thông số dữ liệu cũng như cho biết chính xác tình trạng thực tế công việc đã hoàn thành.

7.6. Ghi lại các vụ tai nạn giao thông và trật tự an ninh nơi công cộng

Công nghệ PROVER có thể được sử dụng để bảo vệ trật tự an ninh nơi công cộng. Với sự trợ giúp của công nghệ này, người sử dụng có thể ghi lại được bằng chứng chân thực nhất của các vụ việc cũng như các vi phạm/hành vi phạm tội với thời gian chính xác diễn ra các sự kiện đó. Dữ liệu sau đó có thể được xác thực bởi các cơ quan chức năng & pháp luật và trở thành bằng chứng trước tòa. Mọi thông tin sẽ về vị trí địa lý, thời gian và khung hình sẽ được xác minh bởi các chuyên gia trong lĩ nh vực kỹ thuật và công nghệ, đảm bảo được sự công bằng của pháp luật.

7.7. Dự án Giáo dục

Trong các dự án giáo dục, công nghệ PROVER có thể được tích hợp để triển khai các tính năng nhận diện người sử dụng từ xa và xác minh các hoạt động của họ trong khi dạy và học. Ngày nay, người dùng trên toàn thế giới đã có thể đăng kí tham dự một khóa học trực tuyến dễ dàng ví dụ như thông qua các trang như Coursera, Udemy, Udacity, edX và nhiều hơn thế nữa. Phần khó khăn nhất của các dị ch vụ này là xác thực được việc các học viên đăng kí học tham dự đúng 100% khả năng và thu nhận đúng lượng kiến thức giảng dạy. hiện nay, trang Coursera đòi hỏi học viên phải tự chụp ảnh của mình khi học bài, tuy nhiên việc này cũng không giúp hệ thống đảm bảo được độ tin cậy của bức hình được chụp. Với PROVER, tạo một thư mục ghi hình, làm bài tập và đảm bảo học viên tham gia, vượt qua kì thi một cách trung thực không phải là vấn đề quá lớn. Công nghệ PROVER cho phép nhà quản lí theo dõi và tránh được tình trạng lừa đảo cũng như xác thực được danh tính học viên tham dự thông qua video gốc xác thực bởi hệ thống. Đoạn phim lưu giữ trong hệ thống cũng có thể được sử dụng để làm bằng chứng xác thực cho các hoạt động khác.

7.8. Mật mã điện tử bảo vệ quyền pháp lý của chủ sở hữu Video

Công nghệ PROVER cho phép người dùng xác minh bản quyền video mà không cần bất cứ cơ quan chức năng nào. Các báo cáo video từ xa, giao dịch được ghi lại, bằng chứng pháp lý, báo cáo điều tra, phỏng vấn, bằng chứng,...

...đều có giá trị về mặt pháp lí bởi hệ thống PROVER sẽ xác thực thời gian và địa điểm video được ghi lại cũng như nhận diện, so sánh với các dữ liệu đã được lưu tương tự trước đó trong hệ thống. Ví dụ như khi nhận các vấn đề thủ tục/hành chính của người dân từ xa, cán bộ nhà nước, các ban bộ có thể sử dụng bản video ghi lại của báo cáo/vấn đề của người dân này để có phương án giải quyết và xử lí thích hợp.

8. Nhóm dự án

Nhóm dự án chúng tôi có kinh nghiệm làm việc nhóm cùng nhau trên 10 năm. Chúng tôi đã triển khai rất nhiều các dự án lớn trong các lĩnh vực về xây dựng hệ thống ghi hình giám sát thông minh, sản phẩm phần cứng và triển khai dịch vụ công nghệ thông tin cho các trung tâm chăm sóc sức khỏe tại 64 quốc gia. Hiện nay, nhóm dự án chúng tôi đang cố gắng bắt kịp xu hướng công nghệ dựa trên nền tảng mã nguồn mở, phân cấp và hệ thống blockchain. Chúng tôi tin rằng dự án PROVER sẽ đóng góp rất nhiều cho sự phát triển của cộng đồng, hệ sinh thái đồng tiền kỹ thuật số bởi hơn ai hết, nhóm dự án chúng tôi luôn nắm bắt và hiểu rõ phương thức tạo ra một sản phẩm chất lượng cao và ứng dụng thiết thực với khách hàng mục tiêu.

8.1. Đội ngũ chính

Nadezhda Nabilskaya, Giám đốc điều hành, Nhà sáng lập

- Cô hoạt động trong lĩ nh vực An ninh bảo vệ thông tin và công nghệ bảo mật từ năm 2010.
- Cô tốt nghiệp chuyên ngành Đào tạo Quản lý thuộc chương trình giảng dạy của Chính phủ Nga
- Quản lý dự án: Phát triển phần mềm - 3 năm, nghiên cứu và phát triển - 5 năm, nghiên cứu ứng dụng - 2 năm.

Alexey Rytikov, Giám đốc Công nghệ

- Có kinh nghiệm trong lĩnh vực phát triển phần mềm bảo mật công nghệ suốt 70 năm. Tham dự với vai trò chủ chốt trong nhiều dự án Nghiên cứu & Phát triển.
- Ông có kinh nghiệm nhiều năm trong các hệ thống kỹ thuật phức tạp trong lĩnh vực giám sát video.

Vyacheslav Voronin, Nhà khoa học trong lĩ nh vực phân tích video

- Phó Giáo sư Đại học Kĩ thuật (Nga).
- Tiến sĩ Khoa học Kĩ thuật.
- Đồng tác giả của quyển sách chuyên đề “Phương pháp & Thuật toán trong tách các tín hiệu hữu ích khỏi tạp âm trong quá trình xử lí tín hiệu rời rạc”.

- Nhà phê bình cho hình ảnh Tạp chí Giao dịch Quốc tế.
- Người xử lý, trong Hội nghị & Tọa đàm Quốc tế về xử lý hình ảnh, Phân tích và Xử lý tín hiệu (ISPA), Hội nghị & Tọa đàm Quốc về Mạch và Hệ thống (ISCAS).
- Người chiến thắng giải cá nhân của cuộc thi người đứng đầu Chính quyền khu vực Rostov (Nga).
- Người đoạt giải “Polzunovskie grants".

Vitoly Suprun, Lập trình viên phát triển ứng dụng di động

- Kinh nghiệm lập trình ứng dụng điện thoại trên 10 năm.
- Lập trình viên của ứng dụng điện thoại ECG Dongle.

Elena Yuferovo, Nhà cố vấn kinh doanh

- ECó 25 năm kinh nghiệm ở vị trí Giám đốc Nhân sự trong các công ty tư vấn & bất động sản và là chuyên gia trong lĩ nh vực Quản lý tổ chức & nhân sự.
- Đồng tác giả của quyển sách Quản trị Nhân sự. Đồng tác giả của quyển sách cho các nhà quản trị “Đối mặt với Nhân viên mới.” M, 2001.

8.2. Ban Cố vấn

Evgeny Shumilov

- Người dẫn đầu trong xu hướng công nghệ Blockchain.
- Nhà sáng lập & Giám đốc điều hành của Emercoin.

Oleg Khovoyko

- Chuyên gia trong ngành Mã hóa và Tài chính.
- Giám đốc công nghệ của Emercoin.

Ilyo Svirin, Nhà Đồng sáng lập

- Tiến sĩ Khoa học Công nghệ
- Doanh nhân Công nghệ.
- Người sáng lập Chuỗi công ty “Nordavind”.
- Lập trình viên công nghệ trong lĩ nh vực hệ thống giám sát video kỹ thuật số, thiết bị cá nhân và dị ch vụ y tế (bao gồm cả dị ch vụ ECG Dongle và dị ch vụ CardioCloud nổi tiếng trên toàn thế giới).
- Ông còn là tác giả của vô số các Ấn phẩm khoa học được phát hành về vấn đề bảo mật thông tin và lý thuyết nền tảng trong lập trình.

9. Lộ trình dự án

Giai đoạn 1
(6 tháng)

Phát triển các thuật toán đọc mã SWYPE dựa trên phân tích đoạn phim. Triển khai "bỏ phiếu" (vote) cho mã Token PROVER trên nền tảng Ethereum.

Phát triển ứng dụng cho điện thoại trên hệ điều hành iOS và Android.

Giai đoạn 2
(6 tháng)

Phát triển thuật toán cho việc xác thực các chuyển động liên tục khi ghi hình dựa trên việc phân tích đoạn phim nhằm tránh tình trạng đoạn phim bị sửa đổi.

Sử dụng mô hình Emercoin cho nhà đầu tư giữ token.

Giai đoạn 3
(6 tháng)

Phát triển thêm một thuật toán xác thực dựa trên việc phân tích bộ cảm biến tích hợp bên trong điện thoại.

Tích hợp thuật toán vào ứng dụng điện thoại.

Giai đoạn 4 ...

Kế hoạch phát triển tiếp theo của dự án PROVER sẽ được thông báo bởi Nhóm dự án.

10. Sổ tay hướng dẫn cho Nhà đầu tư

Để duy trì và gây vốn cho hoạt động của hệ thống PROVER, một đợt gây quỹ, hay còn gọi là Crowdsale sẽ được mở bán. Đợt phát hành này dựa trên nền tảng Ethereum. Trong suốt đợt bán, người tham dự nên mua mã token PROOF với một mức giá cố định:

- Mã PROOF có thể được dùng để thanh toán khi sử dụng dị ch vụ PROVER. Đối với các dịch vụ sử dụng công nghệ PROVER (ví dụ như bảo hiểm xe ô tô), khách hàng của họ có thể sử dụng bất kỳ loại tiền tệ nào như các đồng kỹ thuật số khác hoặc tiền mặt để thanh toán và hệ thống dị ch vụ sẽ chi trả cho chúng tôi bằng đồng PROOF.

- Cho phép dự án được yêu cầu tài trợ. Dự án có thể phát triển dựa trên cấu trúc của PROVER (tạo thêm các thuật toán mới, tạo ra một nền tảng khác), tiếp thị và quảng cáo dự án PROVER, tạo giải pháp cho các ứng dụng dựa trên việc phát triển hệ sinh thái PROVER (phát triển dị ch vụ ứng dụng công nghệ PROVER trong việc xác thực bản quyền video). Số tiền chi cho đề xuất không được lớn hơn tổng số lượng mã PROOF dành cho đối tượng nộp đơn.

- Cho phép người sở hữu Token được bỏ phiếu quýe đị nh về tài chính dự án. Sức mạnh của những người mua PROOF nằm ở số lượng mã token anh ta nắm giữ sẽ biểu thị mức độ ảnh hưởng của quyền biểu quyết.

Để đảm bảo hệ thống hoạt động của PROVER sinh lợi, mã PROOF thu được nhờ dị ch vụ của ứng dụng thông qua hệ thống hợp đồng thông minh sẽ được xoay vòng vốn như sau:

- 70% lượng PROOF thu được trong đó không có video được xác thực nào trị giá dưới 1 PROOF sẽ được nhóm dự án giữ lại để hỗ trợ và phát triển tính năng hoạt động của hệ thống phân cấp.
- 90% lượng PROOF thu được sẽ được lưu trữ dưới sự kiểm soát của hợp đồng thông minh PROOF và phân bổ theo kết quả bỏ phiếu của chủ sở hữu các mã token PROOF từ khi dự án bắt đầu đến khi tiến hành và có lợi nhuận.

10.1. Đợt phát hành Pre-ICO

Hợp đồng thông minh PROOF được phát hành dưới dạng các mã token PROOF, số lượng tối đa có thể được bán trong đợt Pre-ICO được giới hạn ở mức 300 000$, ngay khi đạt được con số trên, việc mở bán ICO sẽ kết thúc. Các mã token được bán với mức giá cố đị nh theo đồng Đô la của Mỹ, 125 PROOF “" 7$, một chủ đầu tư trong đợt phát hành Pre-ICO được thưởng thêm 25% tổng lượng token khi mua so với đợt bán Crowdsale. Giao dị ch được thực hiện bằng cách gửi Ether đến đị a chỉ của hợp đồng thông minh PROOF, sau đó, người đã chuyển Ether nghiễm nhiên sẽ trở thành chủ sở hữu của mã token.

Hãy cẩn thận bởi bạn không nên trả tiền từ những đị a chỉ ví không tương thích với hợp đồng ERC20 hoặc từ một tài khoản giao dị ch trên sàn như Bittrex, Bitfinex,... - điều này có thể dẫn tới việc bạn làm thất lạc Token được nhận khi chúng tôi gửi cho tài khoản ví của bạn.

Tỉ giá giữa đồng Ether và Đô la Mỹ sẽ được chúng tôi cố đị nh tại thời điểm diễn ra đợt bán Pre-ICO và duy trì trong suốt quá trì diễn ra. Đợt Pre-ICO kéo dài 14 ngày kể từ khi bắt đầu. Trong đợt phát hành Pre-ICO, tất cả các mã Token phải được bán, nếu không, chúng tôi sẽ trả lại tiền đầu tư trừ đi khoản phí hoa hồng để chuyển tiền cho giao dị ch và phí GAS.

Mọi tiền vốn từ việc gây quỹ trong đợt Pre-ICO sẽ được chuyển cho nhóm dự án PROVER và sẽ phục vụ cho các công việc sau:

- Phát triển bản thử nghiệm ứng dụng di động trên nền tảng Android tương thích với hợp đồng thông minh PROOF.
- Tiếp thị và quảng bá cho dự án, chuẩn bị cho đợt Crowdsale;
- Phiên dị ch White Paper sang các ngôn ngữ như Trung Quốc, Đức, Pháp, Ý, Tây Ban Nha và Ukraina.

10.2. Crowdsale

Hợp đồng thông minh PROOF được biểu thị dưới dạng mã PROOF, số lượng giới hạn trong đợt gây quỹ của Crowdsale sẽ là 5.200.000$, khi số lượng đạt đến mức này, chúng tôi sẽ ngưng phát hành mã Token. Mã Token sẽ được phát hành theo tỉ giá cố đị nh của đồng Đô la Mỹ, cứ 100 PROOF = 1$. Giao dị ch được thực hiện bằng cách gửi Ether đến đị a chỉ của hợp đồng thông minh PROOF, sau đó, người đã chuyển Ether nghiễm nhiên sẽ trở thành chủ sở hữu của mã token.

Hãy cẩn thận bởi bạn không nên trả tiền từ những đị a chỉ ví không tương thích với hợp đồng ERC20 hoặc từ một tài khoản giao dị ch trên sàn như Bittrex, Bitfinex,... - điều này có thể dẫn tới việc bạn làm thất lạc Token được nhận khi chúng tôi gửi cho tài khoản ví của bạn.

Tỉ giá giữa đồng Ether và Đô la Mỹ sẽ được chúng tôi cố định tại thời điểm diễn ra đợt bán Crowdsale và duy trì trong suốt quá trì diễn ra. Sau khi kết thúc Crowsale một đợt phát hành mã cuối cùng sẽ diễn ra, trong thời gian này, 28% tổng số Token thu được sẽ được phát hành, 19% sẽ được giữ lại cho nhóm dự án, 1% cho các cố vấn và 8% được chuyển cho các nhà đầu tư từ giai đoạn đầu. Crowsale sẽ diễn ra trong vòng 30 ngày kể từ khi bắt đầu.

Ưu đãi dành cho những người đầu tiên tham dự vào đợt bán Crowdsale:

- 115 PROOF = 1$ trong ngày đầu tiên của đợt Crowsale
- 110 PROOF = 1$ trong suốt tuần đầu tiên của đợt Crowsale

Số tiền ban dự án nhận được trong đợt Crowsales sẽ dùng để phát triển nền tảng PROVER và bảo đảm cho các tính năng của nó. Sau đó, chúng tôi sẽ không phát hành thêm mã token nào nữa. Điều kiện thành công của một đợt Crowsales là thu được tối thiểu 3.600.000$, không kể số token bán được từ đợt Pre-ICO. Nếu không thành công, chúng tôi sẽ trả lại tiền đầu tư trừ đi khoản phí hoa hồng để chuyển tiền cho giao dị ch và phí GAS.

Các bước thực hiện được dự án phụ thuộc vào suốt giai đoạn phát hành mã Token và số lượng tính toán cho mỗi đợt. Kết quả mỗi đợt phát hành sẽ góp phần giúp chúng tôi hoàn thành các tính năng sản phẩm, việc bổ sung ở các giai đoạn tiếp theo nhằm giúp chúng tôi bổ sung thêm tính năng cho người tiêu dùng sản phẩm.

10.3. Việc sử dụng vốn đầu tư

Ngay khi hoàn thành đợt bán Crowsale thành công, ngân quỹ 1.500.000$ sẽ được chuyển về ví của người mua trong đợt Pre-sale thông qua hợp đồng thông minh bởi đội ngũ chúng tôi. Phần còn lại của nguồn vốn sẽ được kiểm soát trong hệ thống hợp đồng thông minh PROOF.

Đến cuối mỗi giai đoạn của dự án, nhóm dự án sẽ thông báo kết quả công việc trên trang chủ http://prover.io và tại https://github.com/isvirin/prover, và lên kế hoạch công khai cho giai đoạn tiếp theo cũng như ngân sách tài chính cần thiết. Mọi quyết đị nh sẽ được thông qua bằng hình thức "bầu chọn" kéo dài trong 7 ngày từ ngày nhóm phát triển thông báo.

10.4. Đồng bộ hoá hai mã token PROOF và HMQ

Hai dự án Humaniq và PROVER giữ vai trò hỗ trợ mở rộng hệ sinh thái của nhau. Vì vậy, PROVER và Humaniq được xác thực cùng với nhau dưới dạng đồng PROOF, trong khi đó dự án Humaniq vẫn sử dụng đồng HMQ để giao dị ch với khách hàng của họ.

Do đó, sự gia tăng chi phí của mã token PROOF là do dự án Humaniq giúp PROVER phát triển thêm hệ sinh thái và tạo thêm nhu cầu các mã token PROOF trên thị trường trong cùng một lần phát hành cố định.

Sự gia tăng chi phí của mã token HMQ giúp đảm bảo dự án PROVER nâng cao được tính năng của hệ thống Humaniq, cung cấp thêm giá trị cho khách hàng Humaniq và hỗ trợ hệ thống phát hành mã HMQ vĩ nh viễn.

11. Hướng dẫn người dùng hợp đồng thông minh API

Hợp đồng thông minh PROOF được phát triển bằng cách sử dụng hệ sinh thái Ethereum và các mã Token PROOF được phát hành sử dụng tiêu chuẩn Token ERC20.

11.1. Biểu quyết cho kết quả công việc của nhóm dự án

Đối với chủ sở hữu mã Token dự án, họ có toàn quyền biểu quyết cho kết quả công việc của nhóm dự án, cũng như phê chuẩn kế hoạch tương lai và phân bổ ngân sách cho các giai đoạn tiếp theo. Bỏ phiếu cho nhóm dự án sử dụng StartVoting (uint weiReqFund) một tính năng của hợp đồng thông minh PROOF có thời hạn 7 ngày. Nhóm dự án sẽ nhận được một biến số từ việc đề nghị số tiền vốn cho giai đoạn tiếp theo của dự án. Khi bắt đầu biểu quyết, sự kiện VotingStarted (đồng bộ với VotingStarted (uint weiReqFund) sẽ được thông báo kèm theo thông tin về việc gây quỹ cho giai đoạn tiếp theo.

Để được cung cấp thông tin về cuộc bình chọn đang diễn ra, tìm tính năng VotingInfo để biết thêm tin tức về yêu cầu gây vốn cho giai đoạn sắp tới và thời gian bắt đầu/kết thúc việc biểu quyết. Nếu như việc biểu quyết được hoàn thành hoặc không hoàn thành, tính năng sẽ trả lại kết quả có giá trị bằng 0.

Chức năng bỏ phiếu Bool inSupport chỉ hiển thị với những người nắm giữ mã token của PROOF, có thể được dùng một lần duy nhất cho một phiếu bầu. Việc đếm số phiếu sẽ thực hiện sau khi biểu quyết hoàn thành để tránh tình trạng sử dụng nhiều lần bầu chọn. Mỗi lần chế độ này được gọi thành công, sự kiện Voted(address indexVoter, bool inSupport) sẽ được phát hành kèm theo đó là địa chỉ người đã bỏ phiếu bình chọn và số lượng của phiếu bầu.

Khi muốn bỏ phiếu chấm dứt dự án, sử dụng hàm finishVoting () sau 7 ngày biểu quyết. Tính năng này sẽ tính toán số phiếu, quyền lực của mỗi tài khoản phụ thuộc vào tổng số token họ đang nắm giữ so với token của tất cả những token đang phát hành. Bỏ phiếu tiến hành theo số đông một cách đơn giản. Khi cuộc bỏ phiếu chấm dứt, sự kiện VotingFinished (bool inSupport) sẽ hiện ra cung cấp thông tin kết quả lượt bình chọn. Nếu kết quả khả quan, số quỹ yêu cầu sẽ được chuyển từ hợp đồng thông minh PROOF đến đị a chỉ của nhóm dự án bằng Ether.

11.2. Kiểm tra nơi đặt chứng cứ

Vị trí bằng chứng trong hợp đồng thông minh PROOF được triển khai ở hai giai đoạn. Ở giai đoạn đầu tiên, người dùng gọi tính năng swypeCode () để ngẫu nhiên phát hành mã SWYPE và hiển thị cho người sử dụng. Bên cạnh đó, tính năng lưu trữ swype cod được phát hành trong thời hạn bảo hiểm.

Sau khi người dùng ghi lại đoạn băng (đã có mã SWYPE), người dùng sử dụng tính năng setHash (uint 16 swype, bytes32 hash) để tạo một tham số ngẫu nhiên trước mã SWYPE và hàm băm của đoạn băng được ghi. Chức năng lưu trữ hàm băm trong một chuỗi kết hợp, nơi mà mỗi phân tử đều là một mã swype, thời gian xuất mã swype code, thời gian xuất hàm băm, thời gian video được lưu trữ. Chúng được coi như là những tính năng mang lại lợi nhuận cho dự án cùng với thuế suất, phụ thuộc vào số lượng mã token PROOF mà người sử dụng có.

11.3. Thay đổi nền tảng mã Token

Trong trường hợp thông tin kinh doanh của hợp đồng thông minh bị lỗi hoặc công nghệ kỹ thuật mới có thể hỗ trợ phát triển đáng kể nền tảng của dự án, nhóm kĩ thuật dự án hoàn toàn có quyền thay đổi nền tảng hợp đồng thông minh bằng một hình thức mới. Cùng với đó, để bảo toàn lợi ích của các chủ đầu tư, một hệ thống cơ chế bảo vệ cho chủ sở hữu di chuyển mã Token sẽ được thông báo.

11.4. Tài trợ cho các dự án cộng đồng

Bất kỳ chủ sở hữu mã token PROOF quan tâm đến việc phát triển hoặc quảng bá dự án PROVER có quyền yêu cầu tài trợ. Để mở yêu cầu, người có mã token PROOF dùng hàm deployProject (uint proofReqFund, string urllnfo), các tham số tài trợ theo yêu cầu dự án cũng như đường link đăng tải thông tin dự án kêu gọi. Số tiền chi cho đề xuất không được lớn hơn tổng số lượng mã PROOF dành cho đối tượng nộp đơn nhân cho 1000. Nếu đã tồn tại yêu cầu tương tự trước đó thì không thể mở thêm một yêu cầu mới giống hệt. Bất kể kết quả ra sao, việc mở một yêu cầu mới sau khi việc bầu chọn hoàn thành được chấp nhận.

Nếu yêu cầu được mở ra thành công, sự kiện Deployed (address projectOwner, uint proofReqFund, string urllnfo) sẽ được phát hành và tiến hành bình chọn trong vòng 7 ngày.

Để nhận thông tin về truy vấn bỏ phiếu, sử dụng hàm projectInfo(....) sẽ hồi đáp thông tin yêu cầu gây quỹ, địa chỉ truy cập vào thông tin chi tiết dự án trực tuyến và thời gian trước khi cuộc biểu quyết kết thục. Dù hoàn thành hay không, tính năng này vẫn sẽ trả tham số 0.

Chức năng bầu chọn vote() chỉ được mở cho riêng những người nắm giữ mã token và chỉ được sử dụng một lần trong suốt đợt biểu quyết diễn ra. Việc đếm số phiếu sẽ thực hiện sau khi biểu quyết hoàn thành để tránh tình trạng sử dụng nhiều lần bầu chọn. Mỗi lần chế độ này được gọi thành công, sự kiện Voted(address projectOwner, address voter, bool inSupport) sẽ được phát hành.
Hoàn thành biểu quyết khi sử dụng hàm finishVoting () sau 7 ngày. Tính năng này sẽ tính toán số phiếu, quyền lực của mỗi tài khoản phụ thuộc vào tổng số token họ đang nắm giữ so với token của tất cả những token đang phát hành. Bỏ phiếu tiến hành theo số đông một cách đơn giản. Khi cuộc bỏ phiếu chấm dứt, sự kiện VotingFinished (address projectOwner, boo/ inSupport) sẽ hiện ra cung cấp thông tin kết quả lượt bình chọn. Nếu kết quả khả quan, số quỹ yêu cầu sẽ được chuyển từ hợp đồng thông minh PROOF đến tài khoản của người mở cuộc bầu chọn bằng Ether.

12. Kết Luận

Chiếc điện thoại với máy ảnh ghi hình đầu tiên ra đời khoảng năm 2002, ngay sau đó là kỷ nguyên điện thoại thông minh bắt đầu với chiếc điện thoại iPhone cách đây 10 năm. Đến mãi năm 2005, chúng ta vẫn chưa biết Youtube là gì! Phiên bản đầu tiên của đồng tiền kỹ thuật số Bitcoin ra đời năm 2008, chiếc ví đầu tiên ra đời vào năm 2009 bởi Satoshi Nakamoto. Liệu bạn có thể tưởng tượng được cuộc sống chúng ta ngày nay mà không có chúng?

Số lượng người sử dụng điện thoại trên toàn thế giới hiện nay khoản 4 tỷ người, đến năm 2020, dự tính sẽ là 6 tỷ người. Tất cả chúng ta sẽ giao tiếp thuận tiện bằng tin nhắn, bài viết kèm theo video muôn hình vạn trạng bằng mạng xã hội.

Tất cả điều trên sẽ tạo ra một nền kinh tế mới, chúng ta sẽ mua hàng hoá và dị ch vụ trực tuyến, sử dụng điện thoại thông minh, tiêu dùng đồng tiền kỹ thuật số. Phần lớn các dịch vụ sẽ chuyển sang hình thức trực tuyến và mở rộng quy mô tới hàng triệu người sử dụng. Quá trình này sẽ có thể bị lùi đi bởi nỗi sợ về việc lừa đảo và an ninh mạng.

Công nghệ của chúng tôi tạo sự tin tưởng và là hàng rào cản bước những kẻ gian lận trong thời kì mở cửa và xu thế tiêu dùng trực tuyến, nền kinh tế đồng tiền điện tử cũng như hỗ trợ cho các dị ch dụ Ngân hàng và bảo hiểm, pháp luật và các lĩnh vực kín khác.

Chúng ta cũng có thể chú ý tới lợi ích của các dự án toàn cầu trong việc phát triển hệ thống cộng đồng Blockchain. Hệ thống của PROVER có thể là đầu tàu giúp phổ biến công nghệ Blockchain và đồng tiền kỹ thuật số đến tất cả mọi người trên hành tinh. Điều này sẽ đem lại cho nền kinh tế dựa trên nền tảng Blockchain hàng triệu người sử dụng mới từ khắp mọi nơi trên thế giới, gia tăng cả về số lượng và độ phổ biến của nền kinh tế blockchain.

13. Tham khảo

https://en.wikipedia.org/wiki/Swype
https://en.wikipedia.org/wiki/Insurance_fraud

By the numbers: insurance fraud statistics
Google content ID
Ethereum homepage
Ethereum Request for Comments (ERC) 20
Solidity homepage
How To Learn Solidity: The Ultimate Ethereum Coding Guide
BlockChain Technology Beyond Bitcoin

14. Hợp đồng thông minh PROOF

Dưới đây là một phần của hợp đồng thông minh PROOF: Hợp đồng PROOF là một mã nguồn mở: bạn có thể viết lại, sửa đổi theo các điều khoản của giấy phép GNU, bản gọn của Điều khoản Giấy phép công cộng phát hành bởi Tổ chức phần mềm tự do, hoặc phiên bản 3 của giấy phép, hoặc tuỳ bạn chọn bất cứ phiên bản nào sau này.

Hợp đồng PROOF được công bố với mong muốn nó sẽ hỗ trợ và có ích nhưng không có bất kỳ một sự bảo đảm nào khi sử dụng; cũng không chứa những khả năng về thương mại hoặc tính phù hợp cho một mục đích cụ thể. Xem giấy phép công cộng GNU để có thêm thông tin chi tiết.

Nếu không, xem tại đây: <http://www.gnu.org/licenses/>.

```
pragma solidity "0.4.0;

contract owned {

    address public owner;

    function owned() payable {
        owner = msg.sender;
    }

    modifier onlyOwner {
        require(owner == msg.sender);
        _;
```

```
    }

    function changeOwner(address _owner) onlyOwner public {
        require(_owner != 0);
        owner = _owner;
    }
}

contract Crowdsale is owned {

    uint256 public totalSupply;
    mapping (address => uint256) public balanceOf;

    uint    public etherPrice;
    address public crowdsaleOwner;
    uint    public totalLimitUSD;
    uint    public minimalSuccessUSD;
    uint    public collectedUSD;

    enum State { Disabled, PreICO, CompletePreICO, Crowdsale, Enabled, Migration }
    event NewState(State state);
    State   public state = State.Disabled;
    uint    public crowdsaleStartTime;
    uint    public crowdsaleFinishTime;

    modifier enabledState {
        require(state == State.Enabled);
        _;
    }

    struct Investor {
        uint256 amountTokens;
        uint    amountETH;
    }
    mapping (address => Investor) public investors;
    mapping (uint => address)     public investorsIter;
    uint                          public numberOfInvestors;

    function () payable {
        require(state == State.PreICO || state == State.Crowdsale);
```

```
    uint256 tokensPerUSD = 0;
    if (state == State.PreICO) {
        tokensPerUSD = 125;
    } else if (state == State.Crowdsale) {
        if (now < crowdsaleStartTime + 1 days) {
            tokensPerUSD = 115;
        } else if (now < crowdsaleStartTime + 1 weeks) {
            tokensPerUSD = 110;
        } else {
            tokensPerUSD = 100;
        }
    }
    if (tokensPerUSD > 0) {
        uint valueETH = msg.value;
        uint valueUSD = valueETH * etherPrice / 1000000000000000000;
        if (collectedUSD + valueUSD > totalLimitUSD) { // don't need so much ether
            valueUSD = totalLimitUSD - collectedUSD;
            valueETH = valueUSD * 1000000000000000000 / etherPrice;
            msg.sender.transfer(msg.value - valueETH);
            collectedUSD = totalLimitUSD; // to be sure!
        }
        uint256 tokens = tokensPerUSD * valueUSD;
        require(balanceOf[msg.sender] + tokens > balanceOf[msg.sender]); // overflow
        require(tokens > 0);

        Investor memory inv = investors[msg.sender];
        if (inv.amountETH == 0) { // new investor
            investorsIter[numberOfInvestors++] = msg.sender;
        }
        inv.amountTokens += tokens;
        inv.amountETH += valueETH;
        investors[msg.sender] = inv;
        totalSupply += tokens;
        collectedUSD += valueUSD;
    }
}

function startTokensSale(address _crowdsaleOwner, uint _etherPrice) public onlyOwner {
    require(state == State.Disabled || state == State.CompletePreICO);
    crowdsaleStartTime = now;
```

```
    crowdsaleOwner = _crowdsaleOwner;
    etherPrice = _etherPrice;
    delete numberOfInvestors;
    delete collectedUSD;
    if (state == State.Disabled) {
        crowdsaleFinishTime = now + 14 days;
        state = State.PreICO;
        totalLimitUSD = 300000;
        minimalSuccessUSD = 300000;
    } else {
        crowdsaleFinishTime = now + 30 days;
        state = State.Crowdsale;
        totalLimitUSD = 5200000;
        minimalSuccessUSD = 3600000;
    }
    NewState(state);
}

function timeToFinishTokensSale() public constant returns(uint t) {
    require(state == State.PreICO || state == State.Crowdsale);
    if (now > crowdsaleFinishTime) {
        t = 0;
    } else {
        t = crowdsaleFinishTime - now;
    }
}

function finishTokensSale(uint _investorsToProcess) public onlyOwner {
    require(state == State.PreICO || state == State.Crowdsale);
    require(now >= crowdsaleFinishTime || collectedUSD >= minimalSuccessUSD);
    if (collectedUSD < minimalSuccessUSD) {
        // Investors can get their ether calling withdrawBack() function
        while (_investorsToProcess > 0 && numberOfInvestors > 0) {
            address addr = investorsIter[--numberOfInvestors];
            Investor memory inv = investors[addr];
            balanceOf[addr] -= inv.amountTokens;
            totalSupply -= inv.amountTokens;
            --_investorsToProcess;
            delete investorsIter[numberOfInvestors];
        }
```

```
        if (numberOfInvestors > 0) {
            return;
        }
        if (state == State.PreICO) {
            state = State.Disabled;
        } else {
            state = State.CompletePreICO;
        }
    } else {
        while (_investorsToProcess > 0 && numberOfInvestors > 0) {
            --numberOfInvestors;
            --_investorsToProcess;
            delete investors[investorsIter[numberOfInvestors]];
            delete investorsIter[numberOfInvestors];
        }
        if (numberOfInvestors > 0) {
            return;
        }
        if (state == State.PreICO) {
            if (!crowdsaleOwner.send(this.balance)) throw;
            state = State.CompletePreICO;
        } else {
            if (!crowdsaleOwner.send(1500000 * 1000000000000000000 / etherPrice)) throw;
            // Create additional tokens for owner (28% of complete totalSupply)
            balanceOf[msg.sender] = totalSupply * 28 / 72;
            totalSupply += totalSupply * 28 / 72;
            state = State.Enabled;
        }
    }
    NewState(state);
}

// This function must be called by token holder in case of crowdsale failed
function withdrawBack() public {
    require(state == State.Disabled || state == State.CompletePreICO);
    uint value = investors[msg.sender].amountETH;
    if (value > 0) {
        delete investors[msg.sender];
        msg.sender.transfer(value);
    }
```

```
    }
}

contract Token is Crowdsale {

    string  public standard    = 'Token 0.1';
    string  public name        = 'PROOF';
    string  public symbol      = "PF";
    uint8   public decimals    = 0;

    modifier onlyTokenHolders {
        require(balanceOf[msg.sender] != 0);
        _;
    }

    mapping (address => mapping (address => uint256)) public allowed;

    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(address indexed owner, address indexed spender, uint256 value);

    function Token() payable Crowdsale() {}

    function transfer(address _to, uint256 _value) public enabledState {
        require(balanceOf[msg.sender] >= _value);
        require(balanceOf[_to] + _value >= balanceOf[_to]); // overflow
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
        Transfer(msg.sender, _to, _value);
    }

    function transferFrom(address _from, address _to, uint256 _value) public {
        require(balanceOf[_from] >= _value);
        require(balanceOf[_to] + _value >= balanceOf[_to]); // overflow
        require(allowed[_from][msg.sender] >= _value);
        balanceOf[_from] -= _value;
        balanceOf[_to] += _value;
        allowed[_from][msg.sender] -= _value;
        Transfer(_from, _to, _value);
    }
```

```
    function approve(address _spender, uint256 _value) public enabledState {
        allowed[msg.sender][_spender] = _value;
        Approval(msg.sender, _spender, _value);
    }

    function allowance(address _owner, address _spender) public constant enabledState
        returns (uint256 remaining) {
        return allowed[_owner][_spender];
    }
}

contract MigrationAgent {
    function migrateFrom(address _from, uint256 _value);
}

contract TokenMigration is Token {

    address public migrationAgent;
    uint256 public totalMigrated;

    event Migrate(address indexed from, address indexed to, uint256 value);

    function TokenMigration() payable Token() {}

    // Migrate _value of tokens to the new token contract
    function migrate(uint256 _value) external {
        require(state == State.Migration);
        require(migrationAgent != 0);
        require(_value != 0);
        require(_value <= balanceOf[msg.sender]);
        balanceOf[msg.sender] -= _value;
        totalSupply -= _value;
        totalMigrated += _value;
        MigrationAgent(migrationAgent).migrateFrom(msg.sender, _value);
        Migrate(msg.sender, migrationAgent, _value);
    }

    function setMigrationAgent(address _agent) external onlyOwner {
        require(migrationAgent == 0);
        migrationAgent = _agent;
```

```
        state = State.Migration;
    }
}

contract ProofTeamVote is TokenMigration {

    function ProofTeamVote() payable TokenMigration() {}

    event VotingStarted(uint weiReqFund);
    event Voted(address indexed voter, bool inSupport);
    event VotingFinished(bool inSupport);

    struct Vote {
        bool inSupport;
        bool voted;
    }

    uint public weiReqFund;
    uint public votingDeadline;
    uint public numberOfVotes;
    uint public yea;
    uint public nay;
    mapping (address => Vote) public votes;
    mapping (uint => address) public votesIter;

    function startVoting(uint _weiReqFund) public enabledState onlyOwner {
        require(weiReqFund == 0 && _weiReqFund > 0 && _weiReqFund <= this.balance);
        weiReqFund = _weiReqFund;
        votingDeadline = now + 7 days;
        VotingStarted(_weiReqFund);
    }

    function votingInfo() public constant enabledState
        returns(uint _weiReqFund, uint _timeToFinish) {
        _weiReqFund = weiReqFund;
        if (votingDeadline <= now) {
            _timeToFinish = 0;
        } else {
            _timeToFinish = votingDeadline - now;
        }
```

```
}

function vote(bool _inSupport) public onlyTokenHolders enabledState
    returns (uint voteId) {
    require(votes[msg.sender].voted != true);
    require(votingDeadline > now);
    voteId = numberOfVotes++;
    votesIter[voteId] = msg.sender;
    votes[msg.sender] = Vote({inSupport: _inSupport, voted: true});
    Voted(msg.sender, _inSupport);
    return voteId;
}

function finishVoting(uint _votesToProcess) public enabledState onlyOwner
    returns (bool _inSupport) {
    require(now >= votingDeadline && weiReqFund <= this.balance);

    while (_votesToProcess > 0 && numberOfVotes > 0) {
        address voter = votesIter[--numberOfVotes];
        Vote memory v = votes[voter];
        uint voteWeight = balanceOf[voter];
        if (v.inSupport) {
            yea += voteWeight;
        } else {
            nay += voteWeight;
        }
        delete votes[voter];
        delete votesIter[numberOfVotes];
        --_votesToProcess;
    }
    if (numberOfVotes > 0) {
        _inSupport = false;
        return;
    }

    _inSupport = (yea > nay);

    if (_inSupport) {
        if (migrationAgent == 0) {
            if (!owner.send(weiReqFund)) throw;
```

```
        } else {
            if (!migrationAgent.send(weiReqFund)) throw;
        }
    }

    VotingFinished(_inSupport);
    delete weiReqFund;
    delete votingDeadline;
    delete numberOfVotes;
    delete yea;
    delete nay;
  }
}

contract ProofPublicVote is ProofTeamVote {

  function ProofPublicVote() payable ProofTeamVote() {}

  event Deployed(address indexed projectOwner, uint proofReqFund, string urlInfo);
  event Voted(address indexed projectOwner, address indexed voter, bool inSupport);
  event VotingFinished(address indexed projectOwner, bool inSupport);

  struct Project {
    uint   proofReqFund;
    string urlInfo;
    uint   votingDeadline;
    uint   numberOfVotes;
    uint   yea;
    uint   nay;
    mapping (address => Vote) votes;
    mapping (uint => address) votesIter;
  }
  mapping (address => Project) public projects;

  function deployProject(uint _proofReqFund, string _urlInfo) public onlyTokenHolders
enabledStote {
    require(_proofReqFund > 0 && _proofReqFund <= balonceOf[this]);
    require(_proofReqFund <= bolanceOf[msg.sender] * 1000);
    require(projects[msg.sender].proofReqFund == 0);
    projects[msg.sender].proofReqFund = _proofReqFund;
```

```
    projects[msg.sender].urlInfo = _urlInfo;
    projects[msg.sender].votingDeadline = now + 7 days;
    Deployed(msg.sender, _proofReqFund, _urlInfo);
}

function projectInfo(address _projectOwner) enabledState public
    returns(uint _proofReqFund, string _urlInfo, uint _timeToFinish) {
    _proofReqFund = projects[_projectOwner].proofReqFund;
    _urlInfo = projects[_projectOwner].urlInfo;
    if (projects[_projectOwner].votingDeadline <= now) {
        _timeToFinish = 0;
    } else {
        _timeToFinish = projects[_projectOwner].votingDeadline - now;
    }
}

function vote(address _projectOwner, bool _inSupport) public onlyTokenHolders enabledState
    returns (uint voteId) {
    Project storage p = projects[_projectOwner];
    require(p.votes[msg.sender].voted != true);
    require(p.votingDeadline > now);
    voteId = p.numberOfVotes++;
    p.votesIter[voteId] = msg.sender;
    p.votes[msg.sender] = Vote({inSupport: _inSupport, voted: true});
    projects[_projectOwner] = p;
    Voted(_projectOwner, msg.sender, _inSupport);
    return voteId;
}

function finishVoting(address _projectOwner, uint _votesToProcess) public enabledState
    returns (bool _inSupport) {
    Project storage p = projects[_projectOwner];
    require(now >= p.votingDeadline && p.proofReqFund <= balanceOf[this]);

    while (_votesToProcess > 0 && p.numberOfVotes > 0) {
        address voter = p.votesIter[--p.numberOfVotes];
        Vote memory v = p.votes[voter];
        uint voteWeight = balanceOf[voter];
        if (v.inSupport) {
            p.yea += voteWeight;
```

```
            } else {
                p.nay += voteWeight;
            }
            delete p.votesIter[p.numberOfVotes];
            delete p.votes[voter];
            --_votesToProcess;
        }
        if (p.numberOfVotes > 0) {
            projects[_projectOwner] = p;
            _inSupport = false;
            return;
        }

        _inSupport = (p.yea > p.nay);

        if (_inSupport) {
            transfer(_projectOwner, p.proofReqFund);
        }

        VotingFinished(_projectOwner, _inSupport);
        delete projects[_projectOwner];
    }
}

contract Proof is ProofPublicVote {

    struct Swype {
        uint16  swype;
        uint    timestampSwype;
    }

    struct Video {
        uint16  swype;
        uint    timestampSwype;
        uint    timestampHash;
        address owner;
    }

    mapping (address => Swype) public swypes;
    mapping (bytes32 => Video) public videos;
```

```
    uint priceInTokens;
    uint teamFee;

    function Proof() payable ProofPublicVote() {}

    function setPrice(uint _priceInTokens) public onlyOwner {
        require(_priceInTokens >= 2);
        teamFee = _priceInTokens / 10;
        if (teamFee == 0) {
            teamFee = 1;
        }
        priceInTokens = _priceInTokens - teamFee;
    }

    function swypeCode() public enabledState returns (uint16 _swype) {
        bytes32 blockHash = block.blockhash(block.number - 1);
        bytes32 shaTemp = sha3(msg.sender, blockHash);
        _swype = uint16(uint256(shaTemp) % 65536);
        swypes[msg.sender] = Swype({swype: _swype, timestampSwype: now});
    }

    function setHash(uint16 _swype, bytes32 _hash) public enabledState {
        require(balanceOf[msg.sender] >= priceInTokens);
        require(swypes[msg.sender].timestampSwype != 0);
        require(swypes[msg.sender].swype == _swype);
        transfer(owner, teamFee);
        transfer(this, priceInTokens);
        videos[_hash] = Video({swype: _swype,
timestampSwype:swypes[msg.sender].timestampSwype,
            timestampHash: now, owner: msg.sender});
        delete swypes[msg.sender];
    } }
```