Backing Rich Credentials with a Blockchain PKI*

Karen Lewison and Francisco Corella

October 24, 2016

Abstract

This is the second of a series of papers describing the results of a project whose goal was to identify five remote identity proofing solutions that can be used as alternatives to knowledge-based verification. This paper describes the second solution, which makes use of a rich credential adapted for use on a blockchain and backed by a blockchain PKI. A rich credential, also used in Solution 1, allows the subject to identify him/herself to a remote verifier with which the subject has no prior relationship by presenting verification factors including possession of a private key, knowledge of a password, and possession of one or more biometric features, with selective disclosure of attributes and selective presentation of verification factors. In Solution 2 the issuer is a bank and the biometric verification factor is speaker recognition, which can be combined with face recognition to defeat voice morphing. The paper describes in detail the concept of a blockchain PKI, and shows that it has remarkable advantages over a traditional PKI, notably the fact that revocation checking is performed on the verifier's local copy of the blockchain without requiring CRLs or OCSP.

Contents

*Patent pending.

List of Figures

List of Tables

1 Introduction

This is the second of a series of papers on the results of our research project on *remote identity proofing* [1]. See also a series of posts in the Pomcor blog at `https://pomcor.com/blog/` that summarize and discuss the results.

The goal of the project was to identify five remote identity proofing solutions that can be used as alternatives to knowledge-based verification. We have identified five solutions, and this paper describes Solution 2. Like Solution 1 [2], Solution 2 provides three-factor verification of the identity of a subject to a verifier who has no prior relationship with the subject. It does

so using a variation of the rich credential concept of Solution 1, adapted to take advantage of a blockchain for issuance and verification. It differs from Solution 1 in that the issuer is a bank instead of a DMV, and the biometric modality is speaker recognition, a.k.a. voice recognition, instead of face recognition.

Banks are required to know their customers, and are therefore well placed to provide identity services. In fact, in some countries, banks do provide identity services for the population at large [3, 4]. It would make good business sense for banks to provide such services in the United States, thereby filling a glaring need and tapping an additional source of revenue.

An obstacle to banks becoming identity sources in the US is the traditional resistance of the financial industry to technological innovation. That resistance, however, has been overcome by one particular technology, the blockchain [5, 6, 7]. This suggests that banks may be receptive to the idea of providing identity services if the technology used to provide such services is based on the blockchain. It turns out that a blockchain with *internal*, a.k.a. *on-chain* storage has remarkable advantages for implementing a public key infrastructure (PKI) as explained below in Section 3.4. And a rich credential, just like a traditional public key certificate, needs to be backed by a PKI in broad deployments.

Speaker recognition is popular with banks for routine customer authentication [8, 9, 10, 11]. Therefore a bank may already have voiceprints of its customers before it starts providing identity services. And text-independent speaker recognition lends itself to effective detection of presentation attacks (a.k.a. spoofing attacks) by asking the subject to read prompted text.

2 Overview of Solution 2

All this motivates our second remote identity proofing solution, which can be summarized as follows:

- The identity source is a bank and the subject is an existing bank customer. Before issuing a credential, the bank has acquired a voiceprint of the subject, which it routinely uses to authenticate the customer in the normal course of business. In addition to banking services, the bank also provides an identity service. To take advantage of that service, the subject asks the bank to issue a credential to be used for remote identity proofing.

- The bank issues a *rich blockchain credential* consisting of a private key, a secret salt and a *rich blockchain certificate*, which is identical to a rich certificate as used in Solution 1 and defined in [2], except that it is not signed. The bank computes a hash h of the public key, the metadata and the root label of the typed hash tree of the rich certificate, which we shall call the *certificate hash*, as in Solution 1. But instead of signing the certificate hash, it stores a key-value pair $(h, 1)$ in a blockchain store that it controls, as described below. If the certificate is compromised, the bank revokes it by placing the same key-value pair $(h, 1)$ in another blockchain store that it controls.

- The bank stores the rich blockchain credential in the HTML5 local storage of the subject's browser as in Solution 1. The credential issuance protocol is executed over a TLS connection from the subject's browser to the bank as in Solution 1, except that no security code is used for authentication of the subject. Instead, the subject authenticates by logging in to the bank's web site as a bank customer.

- The rich blockchain credential provides three-factor verification where the biometric factor is speaker recognition. The bank assigns the subject's voiceprint as the label of the biometric template node of a non-revocable biometric modality subtree of the typed hash tree of the rich certificate.

- The subject presents the rich blockchain credential as it would present a Solution-1 rich credential providing the same verification factors, by submitting the rich blockchain certificate and a salted hash of the credential password, and proving knowledge of the private key. If the issuer is not generally known, the rich blockchain certificate is backed by a chain of certificates of certificate authorities (CAs). (The use of the word "chain" when referring to a chain of certificates is unrelated to its use in "blockchain".) The CA certificates are *plain blockchain certificates*, described below.

- The verifier computes the certificate hash as in Solution 1, verifies that it is present in a blockchain store controlled by the credential issuer, and validates the CA certificate chain as described below in Section 3.3. Then it launches a native app as in Solution 1. The app prompts the subject to submit an audio stream of him/herself reading prompted text

selected or generated at random with high entropy. The verifier uses speech recognition to verify that the text being read is the prompted text, while using speaker recognition to match the voice to a voiceprint in the certificate.

It should be noted that the use of the blockchain is limited to storing a cryptographic hash when a credential is issued or revoked, and to allowing the verifier to check for revocation on its own copy of the blockchain. Credentials are not seen on the blockchain, and presentation of a credential does not result in any blockchain activity.

3 Blockchain implementation of a PKI

3.1 Traditional implementation of a PKI

A traditional PKI comprises public key certificates owned by subjects, including end-subjects and certificate authorities (CAs). Each subject has a key pair pertaining to a public key cryptosystem, which must be a digital signature cryptosystem such as ECDSA, DSA or RSA if the subject is a CA. The public key certificate of a subject binds the public key of the subject to certificate metadata and *asserted data* such as attributes of the subject. This binding is accomplished by including in the certificate the public key of the subject, the certificate metadata, the asserted data, and a signature computed by a CA. In a traditional (X.509) public key certificate, the signature is applied to a hash of a one-to-one encoding of the public key, the metadata and the asserted data, using ASN.1 DER encoding [12, Section 4.1]. A rich certificate, as defined in [2], is a public key certificate, but one in which the asserted data is structured as a typed hash tree, and the signature is applied to a hash of a one-to-one encoding of the public key, the metadata, and the root label of the typed hash tree, allowing for selective disclosure of attributes and selective presentation of verification factors when the rich certificate is presented. In both cases we shall refer to the hash that is signed by the CA as the *certificate hash*.

The metadata in a public key certificate typically comprises a version number, a validity period, a serial number, a URL that provides revocation information, an identifier that identifies the digital signature cryptosystem used to sign the certificate, and an identifier that identifies the CA that issued the certificate.

In a PKI with intermediate CAs, an end-subject certificate is the first element of a chain of certificates, where each certificate but the first is a CA certificate, each certificate but the last is verifiable with the public key in the next certificate, and the last certificate is verifiable with the public key of a root CA, which is assumed to be generally known.

3.2 Abstract blockchain with on-chain storage

The concept of a blockchain originated with the invention of the Bitcoin cryptocurrency in 2008 [13]. Many other cryptocurrencies have been launched since then [14], with a wide variety of blockchains [15]. The Ethereum blockchain [16, 17] features a Turing-complete scripting language that can be used to implement smart contracts. For our purposes, however, the important feature of Ethereum is the ability to store and manage data within the blockchain. This feature makes it possible to implement a PKI.

Here we shall abstract away from the complexities of the Ethereum blockchain and define a minimal set of features of a *blockchain with on-chain storage* that make it possible to implement a PKI. Ethereum emulates those features as explained below. Other blockchains could emulate them using different mechanisms.

In our abstract concept of a blockchain with on-chain storage there are *blockchain stores*, each containing a collection of key-value pairs and being controlled by a *key pair* pertaining to a digital signature cryptosystem. (The word *key* has its database meaning in the term *key-value pair* and its unrelated cryptographic meaning in the term *key pair*.) A blockchain store is world-readable, but writing to it requires knowledge of the private key component of the *controlling key pair* (the *controlling private key*). An entity such as a CA that knows the controlling private key has the capability of adding to the store a key-value pair comprising a key not present in the store and an arbitrary value.

These features are emulated on the Ethereum blockchain as follows.

Emulation by the Ethereum blockchain

There are two kinds of accounts in Ethereum, distinguished by whether they have associated Ethereum Virtual Machine (EVM) code. The Ethereum white paper refers to an account without associated code as an *externally owned account*, and an account with associated code as a *contract account*.

A contract account has a store of key-value pairs accessible to the EVM code in the account. EVM code is invoked by a *message call*, which is a remote procedure call specified in a message contained in a blockchain transaction.

An entity such as, for our purposes, a CA generates a key pair comprising a private key and a public key, and uses the private key to sign a blockchain transaction that creates an externally owned account. The externally owned account has an address, which is a hash of the public key. Then the entity signs a transaction that sends a message originating from the externally owned account which results in the creation of a contract account. The message specifies the EVM code of the contract account, which is immutable after the contract account has been created [17, p. 3, col. 2, codeHash paragraph]. The contract account has an address, which is a hash of the address of the externally owned account and a nonce or transaction counter [17, p. 8, eq. 82].

The store of the contract account emulates an abstract blockchain store, and the key pair generated by the entity emulates the controlling key pair. To add a key-value pair to the store, the entity uses its private key to sign a transaction containing a message call from the externally owned account to the contract account, instructing the EVM code in the contract account to store the key-value pair.

The EVM code can determine the externally owned account from which a message originates either directly or via a chain of message calls to several contract accounts [17, p. 24, 0x32 ORIGIN]. The code of the contract account is written so that it provides universal read access to the store but only allows write access to the externally owned account created by the entity. And knowledge of the entity's private key is needed to write messages originating from the externally owned account created by the entity, because such a message must be contained in a blockchain transaction signed with a private key whose associated public key hashes to the address of the account. Therefore the private key does emulate the controlling private key of the abstract blockchain store, and only the entity can write to the store.

3.3 Implementation of a PKI on a blockchain with on-chain storage

A blockchain with on-chain storage having the features specified above in Section 3.2 supports the implementation of a *blockchain PKI*. A blockchain

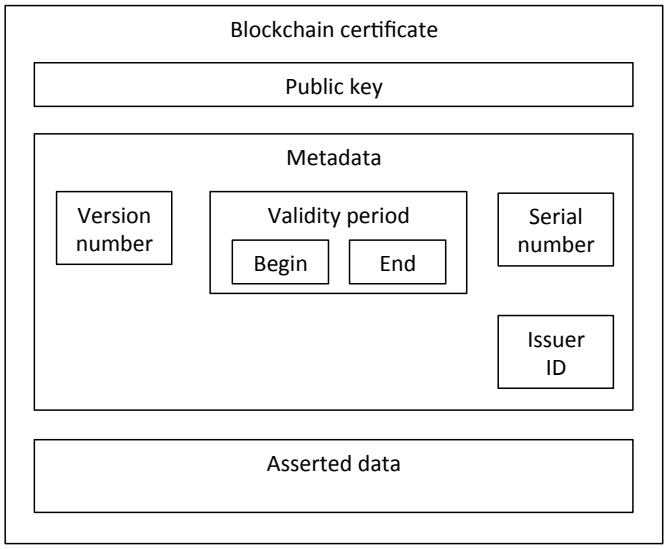

Figure 1: Blockchain certificate

PKI provides the same functionality as a traditional PKI, but has remarkable practical advantages, described below in Section 3.4. A blockchain PKI uses *blockchain certificates*, which are unsigned. Here we shall refer to traditional public key certificates such as X.509 certificates, as well as to the rich certificates of Solution 1, as *signed certificates*, to distinguish them from blockchain certificates. We shall refer to the certificates of Solution 1 as *rich signed certificates* and to traditional public key certificates as *plain signed certificates*.

A *blockchain certificate*, illustrated in Figure 1, comprises a public key, metadata, and asserted data, but no signature. Besides the absence of signature, a blockchain certificate differs from a signed certificate, shown in Figure 2, in that the asserted data does not include a signature cryptosystem ID, since there is no signature in the certificate, nor the URL of a revocation checking service.

Just as there are rich and plain signed certificates, there are rich and plain blockchain certificates. In a rich blockchain certificate, shown in Figure 3, the asserted data is a typed hash tree as in Solution 1, and the certificate hash is computed on the public key, the metadata, and the root label of the typed hash tree. By contrast in a plain blockchain certificate, the certificate hash is computed on the public key, the metadata and the asserted data.

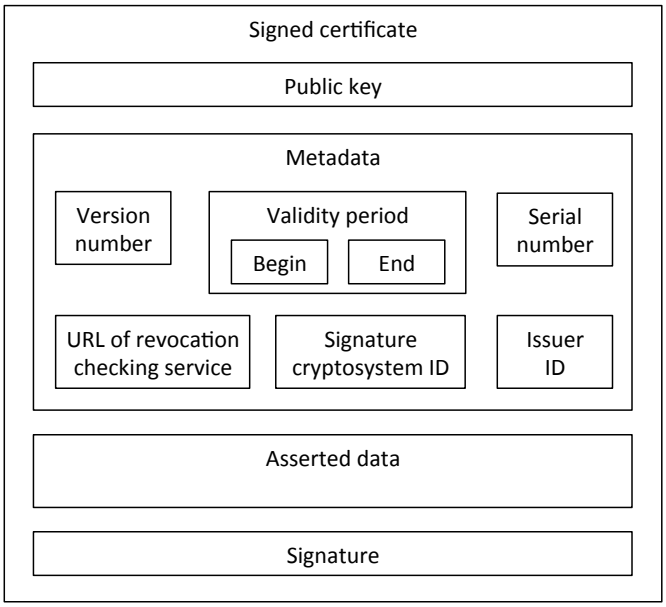

Figure 2: Signed certificate

	Rich	Plain
Blockchain	End-subject certificate in Solution 2	CA certificate in Solution 2
Signed	End-subject certificate in Solution 1	CA certificate in Solution 1

Table 1: Certificate usage in solutions 1 and 2

A rich blockchain certificate is a component of a *rich blockchain credential*, which also comprises as other components the private key associated with the public key in the certificate, and a secret salt as in Solution 1. A plain blockchain certificate, on the other hand, may be viewed as a component of a plain blockchain credential whose only other component is a private key.

In a blockchain PKI that backs rich credentials as in Solution 2, end-subject certificates are rich blockchain certificates, while CA certificates are plain blockchain certificates. Similarly, in a traditional PKI that backs rich credentials as in Solution 1, end-subject certificates are rich signed certificates, while CA certificates are plain signed certificates. Certificate usage in solutions 1 and 2 is summarized in Table 1.

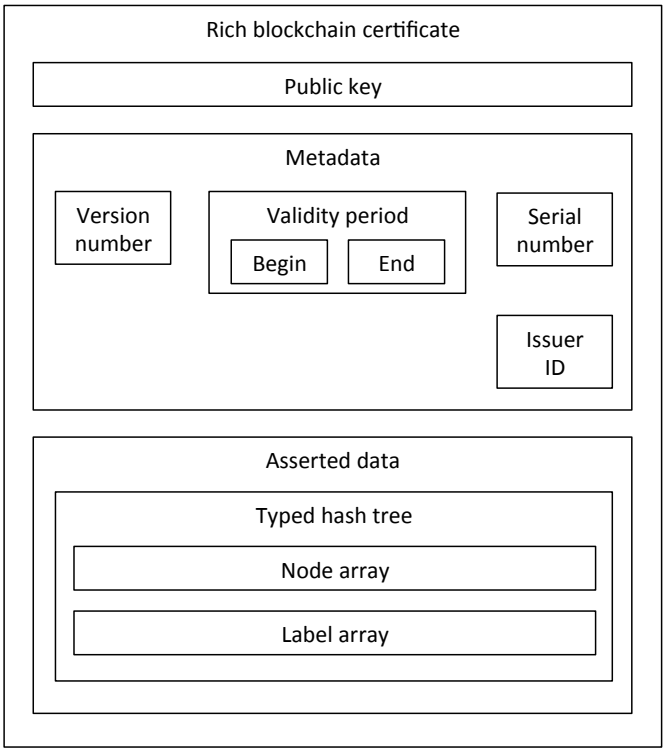

Figure 3: Rich blockchain certificate

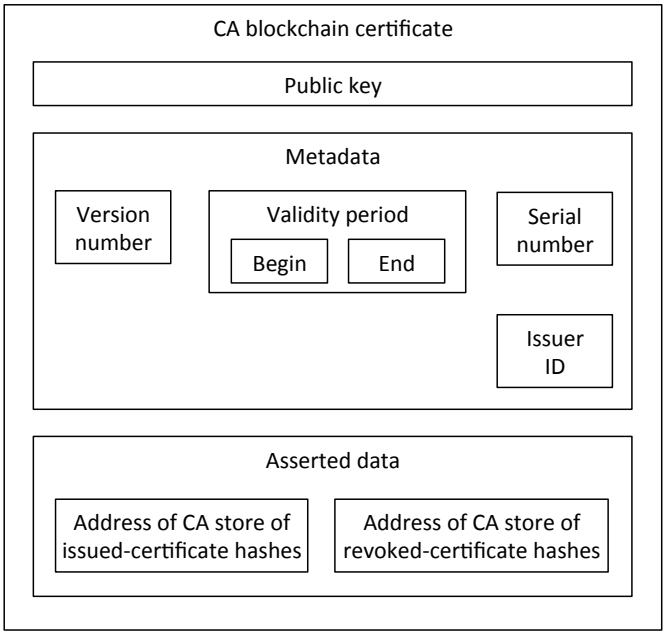

Figure 4: CA blockchain certificate

To validate an end-subject blockchain certificate the verifier checks that the certificate hash is present in the issuer's certificate store and not present in the issuer's revoked certificate store.

In a deployment where there is only one issuer, the blockchain addresses of those stores are generally known. In a narrow deployment where there are multiple issuers but all issuers are known to all verifiers, the verifier identifies the issuer by the issuer ID component of the metadata, and knows the addresses of the stores of the identified issuer.

In a broad deployment where the verifier may not know the issuer, the subject's rich blockchain certificate is backed by a chain of CA certificates. The issuer is a CA, and the chain begins with the issuer's certificate. Each CA certificate is a plain blockchain certificate, which contains in its asserted data the blockchain addresses of the CA's issued-certificate store, where the CA stores the pair $(h, 1)$ for each certificate hash h of a certificate that it issues, and the CA's revoked-certificate store, where it stores the pair $(h, 1)$ for each certificate hash h of a certificate that it has revoked, as shown in Figure 4. The verifier checks that the certificate hash of the end-subject certificate is present in the issued-certificate store and not present in the

revoked-certificate store of the issuer, using the addresses found in the first certificate of the chain. Then the verifier validates each CA certificate in the chain, verifying that the certificate hash is present in the issued-certificate store and not present in the revoked-certificate store of the higher level CA that issued the certificate. For all CA certificates in the chain but the last, the verifier finds the addresses of those stores in the asserted data of the next certificate in the chain. The last certificate in the chain is issued by a root CA, whose issued-certificate and revoked-certificate stores have generally known blockchain addresses.

It should be noted that there is a correspondence between validation of a CA certificate chain in a blockchain PKI and validation of a CA certificate chain in a traditional PKI. Signature verification in a traditional PKI corresponds to checking the presence of the certificate hash in a blockchain store, and the public key used to verify the signature in a traditional PKI corresponds to the address of the blockchain store. An important improvement in the validation method of the blockchain PKI is that the revocation check is performed by checking the absence of the same certificate hash in another blockchain store, obviating the need for a separate revocation process. The revocation check could be eliminated altogether by simply removing the certificate hash from the issued-certificate store when the certificate is revoked, in a blockchain that supports the removal of a key-pair from a blockchain store.

3.4 Advantages of a blockchain implementation of a PKI

A blockchain PKI has the following advantages over a traditional PKI.

First, certificates are not signed. This means that they are shorter, which reduces the time it takes to transmit a certificate backed by a CA certificate chain.

Second, validation of a certificate and its CA certificate chain is trivial. A blockchain being a *distributed ledger*, the verifier has a local copy of the entire blockchain and looks up hashes of certificates in blockchain stores in the local copy, without network access. No signatures need to be verified.

A blockchain PKI solves a longstanding problem of traditional PKIs by not requiring the use of a service that issues certificate revocation lists (CRLs) or responds to online certificate status protocol (OCSP) queries. CRLs can

get very big. They must be stored by the verifier and updated over the network on a regular schedule. OCSP checks add network latency to certificate validation, and leak the information that the subject is presenting the certificate to the verifier, destroying the unobservability feature of cryptographic credentials. (See our paper [18] on the privacy features of various kinds of credentials and authentication methods.) Too often, if the revocation checking service is unavailable, verifiers skip revocation altogether.

It should be noted that a blockchain PKI can be used to back plain blockchain certificates just as well as rich blockchain certificates, and both use cases benefit from the above advantages of a blockchain PKI.

4 Security analysis

4.1 Threat model and adversarial capabilities

The general threat model of [2, Section 8.1] and the specific adversarial capabilities against Solution 1 are applicable to Solution 2, *mutatis mutandis*, except that there is no vulnerability to the capture of a security code, since no such code is used by the subject to retrieve a rich blockchain credential from the issuing bank. Instead, solution 2 takes advantage of the preexisting relationship of the issuing bank with the subject, who is a customer of the bank.

A large coalition of miners who collude to undo transactions is a potential adversary against Solution 2. This is discussed below as Threat 1.

4.2 Threats and mitigations

Threat 1

Recall that the bank that issues a rich blockchain certificate revokes a compromised certificate by creating a transaction that places its hash in a store of hashes of revoked certificates. A large coalition of miners may be able to undo that transaction by creating a fork that does not include the transaction. The subject of the certificate may then be able to successfully present the revoked certificate to a verifier. If the presentation is for the purpose of remote identity proofing, this is a concern if the certificate was issued in error or the subject is no longer entitled to an attribute included in the certificate.

It should be noted that no coalition of miners can issue or revoke certificates, since access to the blockchain stores of hashes of issued certificates and revoked certificates requires the private key of the issuer, as explained above in Section 3.2.

Mitigation of Threat 1

The bank can reduce the risk that a revocation transaction is permanently undone by monitoring the blockchain and repeating the transaction as needed. But it should be noted that the existence of a coalition of miners capable of forking the block chain to undo transactions would be a much larger problem than the possibility of certificate revocations being undone. The existence of such a coalition would put at risk the legitimacy of the entire blockchain.

Threat 2

An adversary impersonates a subject by reading the prompted text and using *voice morphing* to disguise the adversary's voice on-the-fly so that it sounds like the subject's [19].

Mitigation of Threat 2

The bank combines speaker recognition with face recognition, taking advantage of the ability of the typed hash tree of a rich credential to support multiple biometric modalities. To do so, at issuance time the bank acquires a facial image of the subject, and places it in the typed hash tree of the rich blockchain certificate as verification data of a non-revocable biometric modality. The subject is asked to submit an audio-visual stream to the verifier, instead of just an audio stream. The verifier extracts the face from the video channel and matches it against the facial image in the certificate. The verifier also verifies synchrony of the audio and video channels as in Solution 1, by correlating distinguishable visemes against phonemes.

Threat 3

A subject repudiates participation in a remote identity proofing event, or a verifier falsely implicates a subject in an event.

As in Solution 1 [2, Section 8.3, Threat 6], the traditional defense against fraudulent repudiation based on the private key not being shared is not available because the private key is stored in HTML5 local storage that the *same origin policy* of the web makes accessible to the issuer's web application.

Mitigation of Threat 3

The verifier can help adjudicate a repudiation dispute by recording and retaining the audio or audio-visual stream submitted by the subject.

Other threats and mitigations

Threats 1–3 and 5 of Solution 1 [2, Section 8.3] and their mitigations apply to Solution 2 as well.

4.3 Security posture of Solution 2

The security posture of Solution 2 is similar to the security posture of Solution 1 [2, Section 8.4], from which it differs as follows:

- As noted above in the threat model, Solution 2 is not vulnerable to the capture of a security code used to retrieve a credential, since no such code is used. Instead, its security is subordinate to the security of the web account of the subject with the bank, since the credential issuance protocol takes places over a TLS connection established when the subject logs in to the bank's web site as a customer.

- Credential revocation is performed locally using the verifier's copy of the blockchain. It does not require access to a revocation checking service over the Internet. Hence verifiers will not be tempted to skip the revocation check if the revocation checking service is not available, something which is said to occur too often.

- If speaker recognition is performed by itself rather than in combination with face recognition, the biometric verification factor of Solution 2 is vulnerable to voice morphing.

5 Conclusion

This paper has described Solution 2, the second of five remote identity proofing solutions that we have identified as possible alternatives to knowledge-based verification. Like Solution 1 [2], Solution 2 is based on the concept of a *rich credential*, a new kind of cryptographic credential that can be used by a subject to remotely present verification factors to a verifier with which the subject has no prior relationship, including possession of a private key, knowledge of a password, and possession of one or more biometric features, with selective disclosure of attributes and selective presentation of verification factors. In Solution 2 the issuer is a bank and the biometric verification factor is speaker recognition, which can be combined with face recognition to defeat voice morphing.

In Solution 2, however, the rich credential is adapted for issuance and verification on a blockchain, by including a rich blockchain certificate, which is unsigned, instead of a rich signed certificate as in Solution 1. The rich blockchain certificate is asserted on a blockchain, and backed by a blockchain PKI rather than an ordinary PKI. The paper has described in detail the concept of a blockchain PKI, and shown that it has remarkable advantages over a traditional PKI, notably the fact that revocation checking is performed on the verifier's local copy of the blockchain without requiring CRLs or OCSP. These advantages are available both when a blockchain PKI is used to back rich blockchain certificates and when it is used to back traditional certificates similarly adapted for use on the blockchain.

6 Acknowledgments

This paper is the result of a project sponsored by an SBIR Phase I grant from the US Department of Homeland Security. It does not necessarily reflect the position or the policy of the US Government.

We are very grateful to Prof. Phil Windley for many comments on project working documents.

References

[1] Francisco Corella. Pomcor Receives DHS Grant to Look for Alternatives to Knowledge-Based Verification for Remote Identity

Proofing. Pomcor blog post, August 3, 2016. https://pomcor.com/2016/08/03/pomcor-receives-dhs-grant-to-look-for-alternatives-to-knowledge-based-verification-for-remote-identity-proofing/.

[2] Karen Lewison and Francisco Corella. Rich Credentials for Remote Identity Proofing. Pomcor technical report, October 15, 2016. https://pomcor.com/techreports/RichCredentials.pdf.

[3] Finansiell ID-Teknik BID AB. BankID. https://www.bankid.com/en/.

[4] Justin Lee. Norway's BankID selects Encap Security platform for digital ID pilot. June 27, 2016. http://www.biometricupdate.com/201606/norways-bankid-selects-encap-security-platform-for-digital-id-pilot.

[5] Laura Shin. Bitcoin Technology Tested In Trial By 40 Big Banks. Forbes, March 3, 2016. http://www.forbes.com/sites/laurashin/2016/03/03/bitcoin-technology-tested-in-trial-by-40-bigbanks/.

[6] Cade Metz. Why Wall Street Is Embracing the Blockchain—Its Biggest Threat. Wired, February 16, 2016. http://www.wired.com/2016/02/wall-street-is-embracing-the-blockchain-its-biggest-threat/.

[7] Arjun Kharplan and Julia Chatterley. Bank of America is going big on blockchain. CNBC, January 28, 2016. http://www.cnbc.com/2016/01/28/bank-of-america-is-going-big-on-blockchain-plans-to-file-20-patents.html.

[8] Barclays. Banking on the power of speech. https://wealth.barclays.com/en_gb/home/international-banking/insight-research/manage-yourmoney/banking-on-the-power-of-speech.html.

[9] Mark Ryan. Will Voice Recognition Kill Online Banking? March 10, 2016. http://thefinancialbrand.com/57631/online-banking-voice-recognition-authentication/.

[10] Amar Toor. HSBC brings Touch ID and voice recognition to UK banks. February 22, 2016. http://www.theverge.com/2016/2/22/11091876/hsbc-touch-id-voice-recognition-biometric-bank.

[11] Stephen Mayhew. OCBC Bank says voice biometrics for authentication will be available to all retail customers by Q4. May 24, 2016. http://www.biometricupdate.com/201605/ocbc-bank-says-voice-biometrics-for-authentication-will-be-available-to-all-retail-customers-by-q4.

[12] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and CRL Profile, May 2008. http://datatracker.ietf.org/doc/rfc5280/.

[13] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. October 2008. https://bitcoin.org/en/bitcoin-paper.

[14] CoinMarketCap. Crypto-Currency Market Capitalizations. http://coinmarketcap.com/.

[15] Nomura Research Institute. Survey on Blockchain Technologies and Related Services. March 2016. http://www.meti.go.jp/english/press/2016/pdf/0531_01f.pdf.

[16] Ethereum Wiki. Ethereum White Paper. https://github.com/ethereum/wiki/wiki/White-Paper.

[17] Gavin Wood. Ethereum: A Secure Decentralised Generalised Transaction Ledger—Homestead Revision. http://gavwood.com/Paper.pdf.

[18] Francisco Corella and Karen Lewison. Privacy Postures of Authentication Technologies. http://pomcor.com/techreports/PrivacyPosturesMayFirstVersion.pdf.

[19] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. All Your Voices Are Belong to Us: Stealing Voices to Fool Humans and Machines. European Symposium on Research in Computer Security

(ESORICS), September 2015. Available at
https://www.cis.uab.edu/saxena/docs/mss-esorics15.pdf.