TECHNICAL WHITE PAPER

NOK NOK LABS MULTIFACTOR AUTHENTICATION

Any device.
Any application.
Any authenticator.

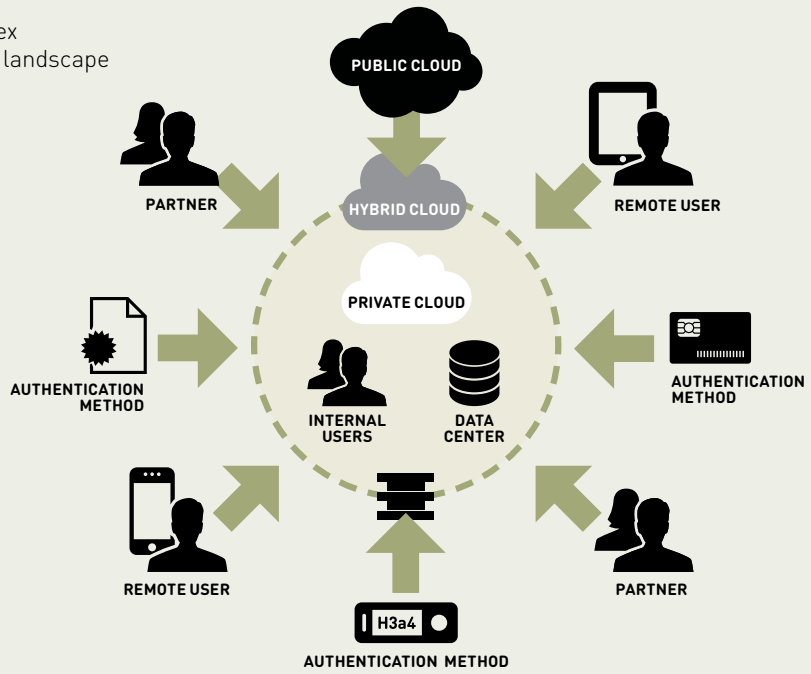
Nok Nok
LABS

TABLE OF CONTENTS

Nok Nok
LABS

INTRODUCTION

Figure 1
Today's complex authentication landscape



Authentication used to be a manageable problem. Users signed in from stationary desktops, used standard devices, and needed access to a limited number of applications. The applications could be co-located at an on-premises datacenter and protected with a firewall, using the network as the perimeter. For authentication purposes, passwords were convenient and sufficient. They were the lowest common denominator—simple, cheap, and usable anywhere.

When certain sensitive applications needed stronger forms of authentication, we used solutions such as hardware tokens and, to a lesser degree, public key infrastructure (PKI). Although such strong authentication methods were cumbersome and expensive, because their use was limited to special use cases in enterprise and online banking, we learned to live with their drawbacks. Not only was authentication simple but also the costs of security breaches were moderate. As the amount of valuable online information was limited, so too was the damage done by a breach. Consequently, security architectures and authentication systems were built to fit a relatively simple and static world.

However, within just a few years, the world changed drastically (Figure 1). Users moved from the standardized world of PCs to the fragmented and diverse world of smartphones and tablets. Users have become mobile, working from remote locations and switching among devices. Applications have become mobile too. No longer are they hosted only at a centralized datacenter; they can be dispersed in the cloud or hosted by partners; they can even be local on mobile devices.

To complicate matters, the information available online has exploded with the growth of applications such as enterprise cloud services, online banking, and social networks. This information is now dispersed across multiple accounts and interlinked, enabling new kinds of attacks and increasing the damage caused by each breach.[1] Even when individual applications may be secure on a standalone basis, the interaction of interlinked user accounts exposes new and unanticipated weaknesses in the security of these individual applications. Breaches have started to exhibit the domino effect, with each breach contributing to a subsequent one.

1. An example of this type of attack is the 2012 hacking of Mat Honan's (Wired magazine) online accounts. http://www.wired.com/gadgetlab/2012/08/apple-amazon-mat-honan-hacking/

THE PROBLEM WITH AUTHENTICATION TODAY

Neither passwords nor strong authentication methods have evolved to meet the needs of users and organizations. As the number of services used by a typical user has multiplied, so too has the number of usernames and passwords the user needs to remember. Users try to manage the problem by using common or easy-to-remember passwords. However, such passwords are more vulnerable to attack and weaken security. In response, online and mobile application providers (called *relying parties*) have increased password complexity by adding requirements for upper case characters, special characters, and numbers. Passwords have become ever more difficult to remember and type, especially on the small and cumbersome keyboards common on mobile devices. This further motivates users to memorize one password and reuse the same password everywhere, perpetuating a vicious cycle of weakened security and increased friction in the user experience. Higher friction frustrates users and increases failed logins, lowering usage and user engagement.

Older forms of strong authentication were not designed to address today's problems either. Diversity in devices, locations, and applications result in a corresponding diversity in authentication use cases. However, most strong authentication systems address only a subset of the use cases required by organizations. For example, a bank may use hardware tokens costing between $10 and $50 to protect high-value accounts. However, these same tokens will not be cost effective for the bank's mobile payments division, which processes an average of only $50 per user per year.

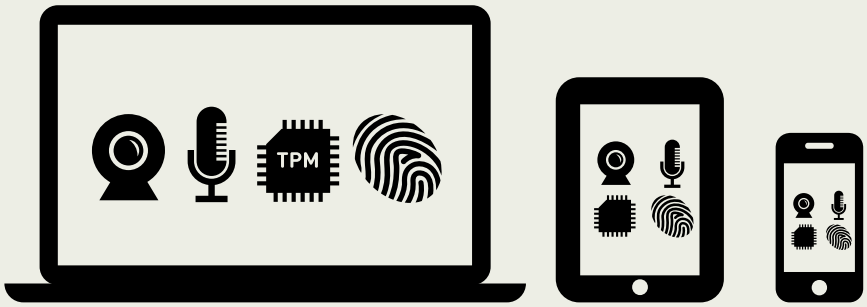
Organizations have coped by deploying multiple authentication stacks, each addressing a handful of use cases. However, these stacks are not integrated, resulting in authentication silos that must be deployed, provisioned, and managed separately. As the number of use cases multiplies, so too does the number of parallel stacks, leading to expensive, complex and unmanageable authentication infrastructures. Neither the piecemeal approach of strong authentication nor the lowest common denominator approach of passwords is addressing the needs of organizations and users today.

NEW POSSIBILITIES

However, not all trends in the world of authentication are bad. Fortunately, many devices already include technologies that could be used to provide user-friendly and inexpensive strong authentication (Figure 2). For example, laptops and desktops have long shipped with Trusted Platform Modules (TPMs) and, in many cases, fingerprint sensors. Smartphones and tablets feature powerful cameras and sensitive microphones that can be used for biometric authentication. Newer mobile devices also include fingerprint sensors. Manufacturers are beginning to equip devices with secure hardware such as Trusted Execution Environments (TEEs) and Secure Elements (SEs) — embedded capabilities that can provide hardware-level protection for sensitive data and operations. Today's devices feature powerful multicore processors and gigabytes of memory, enabling use of computation-intensive authentication algorithms.

Although today's devices have the building blocks for strong authentication, organizations have no way to take advantage of those mechanisms for their online or mobile applications. Smartphones may ship with cameras, but few include face biometric software. Even smartphones that do ship with a biometric capability limit its use to screen unlock. For example, it is not possible for a user to authenticate to Facebook using his smartphone's camera, because the face biometric application is not integrated with Facebook's mobile application. The necessary integrations that would solve the authentication problem have not existed—until now. Recognizing this critical need, Nok Nok Labs has developed the solution.

Figure 2
Authentication mechanisms available on devices



UNIFYING AUTHENTICATION INFRASTRUCTURE

Nok Nok Labs provides the missing software and integrations needed to enable any application to make use of any authentication method present on any device. In effect, the Nok Nok solution enables applications to take advantage of security capabilities already present in billions of devices for strong authentication. It achieves this by plugging these capabilities into a standards-based, end-to-end platform based on the Universal Authentication Framework (UAF) defined by the Fast IDentity Online (FIDO) Alliance.[2]

Nok Nok Labs allows organizations to deploy a single unified system that addresses all their authentication needs. It addresses the user experience disadvantages of passwords by minimizing the need for them. It addresses the fragmented and silo-based nature of today's strong authentication solutions by creating a unified, modular authentication infrastructure that supports any authentication method on any device. Furthermore, it provides future-proofing for organizations, enabling them to easily support new authentication methods as they become available. The Nok Nok Labs solution consists of an authentication client that is installed on the user's device, an authentication server that integrates with the relying party application, and an authentication protocol that allows the client and the server to communicate.

UNIVERSAL AUTHENTICATION FRAMEWORK PROTOCOL

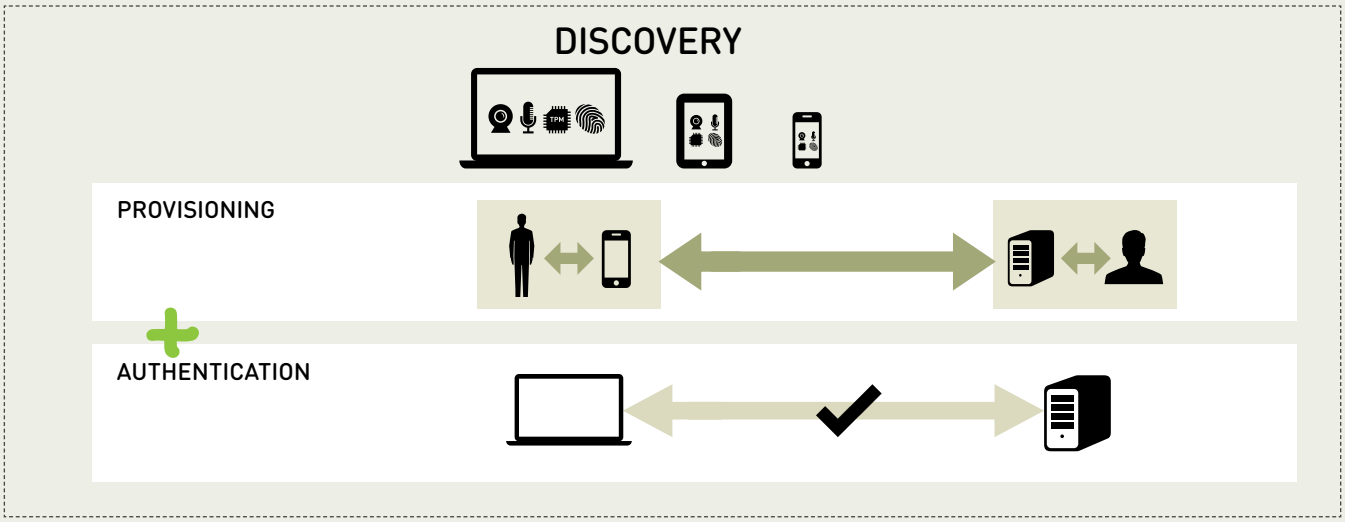
The key to making all this possible is the Universal Authentication Framework (UAF) protocol , a modular and extensible industry standard protocol that enables authentication to take place using virtually any authentication method or authenticator. UAF abstracts away the specifics of each authenticator, achieving interoperability between the authentication server and the authenticator. All communication between the client and the server takes place over a secure Transport Layer Security (TLS) channel.

UAF supports two operations: provisioning, and authentication with a discovery component functioning in both operations (Figure 3).

DISCOVERY
The discovery component is the cornerstone of UAF's ability to integrate the virtually any authenticator with the end-to-end authentication framework. UAF uses a policy-based discovery mechanism to negotiate the appropriate authentication method to employ in an operation. During each operation, the server communicates a policy to the client specifying which authentication methods it is willing to accept. The client presents the policy options available on the device to the user, allowing the user to select his/her preferred options for the relying party. This approach avoids privacy concerns by enabling the usage of the device's capabilities without explicit enumeration of the capabilities to the server.

2. See www.fidoalliance.org.

Figure 3
UAF Operations



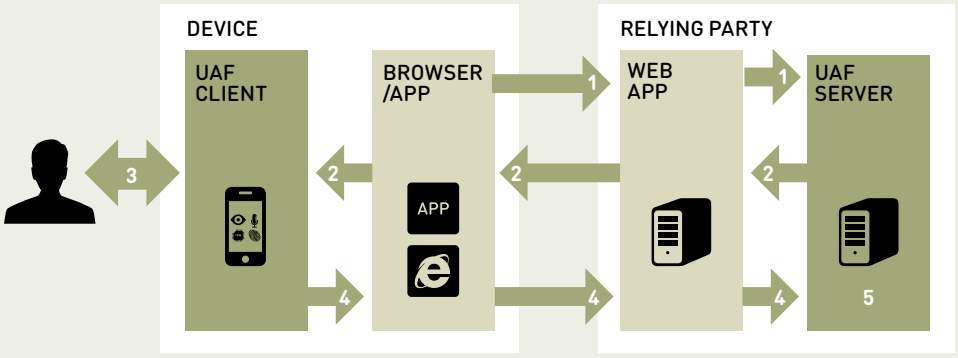
PROVISIONING
During the provisioning operation (Figure 4), the client allows users to enroll their device using one or more authenticators, based on a list of acceptable authenticators provided by the server. The client generates a pair of authentication keys unique to the user, the authenticator, and the relying party. The public key is sent to the server; the private key is stored securely on the client.

As part of this process, the server also verifies that the authenticator is genuine, using a preestablished attestation key. This reduces the risk of an attacker spoofing the authenticator.

Provisioning may require authenticator specific actions from the user. For example, a user may enroll using a fingerprint sensor by swiping her finger several times to register her fingerprint. Once the user is enrolled successfully, the new key pair is generated. For future provisioning operations with other relying parties, the enrolled user would need to swipe her finger once to link her existing biometric data to a new account.

To protect the user's privacy, the biometric data is not sent to the server. Instead, a non-reversible template[3] generated from the data is stored locally in a secure manner and is used during authentication to trigger the unlocking of the private authentication key.

Figure 4
Provisioning in UAF



1 Initiate provisioning step.
2 Send registration request, including registration policy.
3 Enroll user to authenticator(s) selected according to policy and generate new key pair(s).
4 Send registration response, including public key to web app.
5 Validate registration response (including attestation), using trust store.

3. The precise nature of the template will depend on the authenticator used.

AUTHENTICATION
UAF authentication consists of two parts (Figure 5):

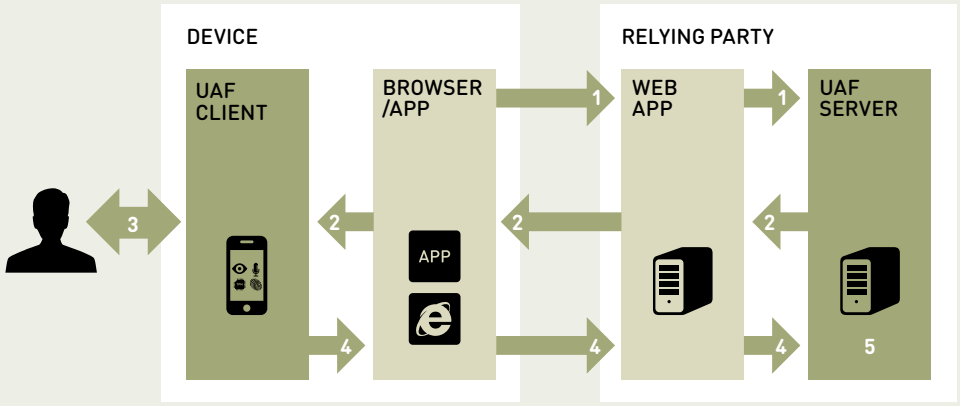
1. Local authentication of the user to the client

2. Authentication of the client to the server

The authentication to the server is performed using a challenge-response protocol. The server generates a challenge and sends it to the client in the authentication request. The server also indicates one or more authenticators it will accept for the session in the request message. The client selects these authenticators and asks the user to perform the actions required by the specific authentication modality. For example, the voice biometric authenticator may instruct the user to provide a voice sample. Once the user is authenticated locally, the client computes a response to the challenge, using the previously provisioned authentication key, and communicates the response back to the server. By verifying the response, the server authenticates that the user has control of the authentication key. Depending on the device capabilities, the computation of the authentication response may be performed in software or with secure hardware.

As part of the authentication step, UAF provides the capability to obtain user confirmation for a transaction using authentication. Transaction details can be presented to the user for review; allowing the user to approve the transaction by authenticating to the device. This approach also allows a relying party to use a second device, such as a mobile device, as a transaction confirmation terminal that is independent from the device used to initiate the transaction.

Figure 5
Authentication in UAF



1 Initiate authentication step.
2 Send authentication page with challenge and authentication policy.
3 Authenticate user using authenticator selected according to policy.
4 Send authentication response to web app.
5 Validate authentication response with public key.

THE S3 AUTHENTICATION SUITE

Nok Nok Lab's (NNL™) S3 Authentication Suite re-imagines authentication to be simple, strong and scalable. The FIDO Ready™ S3 Authentication Suite consists of:

— The Multifactor Authentication Server (MFAS)
— The Multifactor Authentication Client (MFAC) Mobile Edition with support for Android and iOS devices (Also includes the Mobile App SDK and Authenticator Specific Module (ASM) SDK).
— The Multifactor Authentication Client (MFAC) Desktop Edition, with support for Windows 7 and Windows 8.

The S3 Authentication Suite is the only authentication platform to support the full suite of FIDO authentication modes including the passwordless mode (using the Universal Authentication Framework Protocol) and the password augmentation mode (using the Universal Second Factor Protocol). [4]

4. A discussion of the Universal Second Factor Protocol is beyond the scope of this paper.

MULTIFACTOR AUTHENTICATION CLIENT

Multifactor Authentication Client (MFAC) serves as an UAF endpoint, enabling the native capabilities of the device to be used for authentication. MFAC is available on Windows, Android, and iOS devices today and can be preinstalled by the device vendor, bundled with mobile applications, or distributed by the relying party.

During the authentication process, MFAC authenticates the user locally using one or more authenticators as directed by the server. Successful local authentication unlocks a relying party-specific authentication key that is used to authenticate the user to the server. This mechanism has two advantages:

1. Details of the local authentication are abstracted from the server, allowing any authentication method to be used.

2. Sensitive biometric information is not sent to the server, thereby protecting the user's privacy.

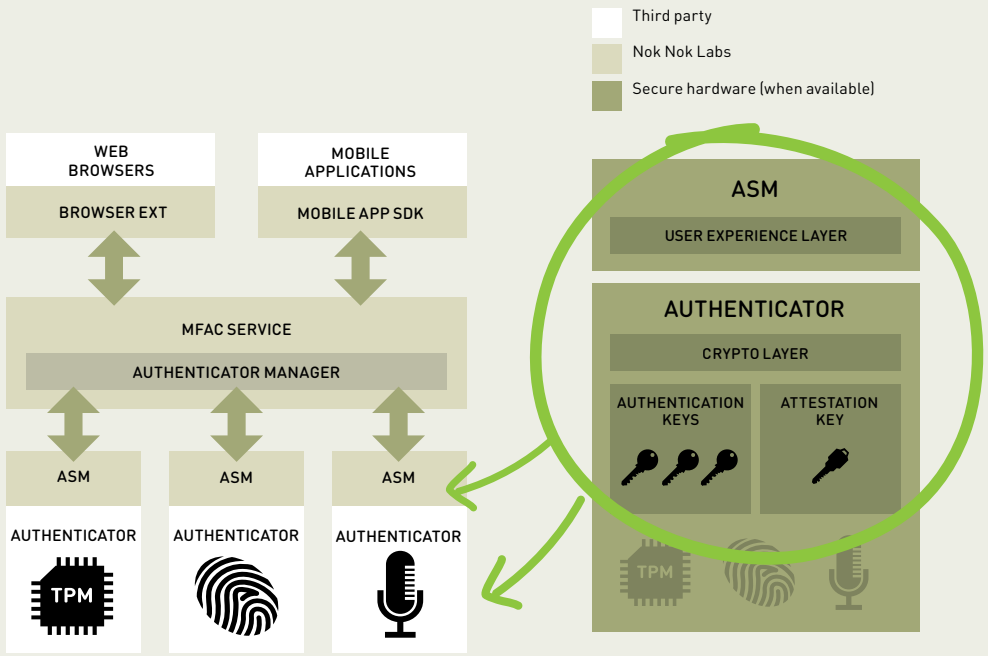
MFAC features a pluggable architecture and is designed to be highly extensible (Figure 6). This design allows for integration with web and mobile applications as well as with any kind of authenticator hardware. Web applications communicate with MFAC using browser extensions, which form a conduit between the application and MFAC. Native mobile applications integrate with MFAC using the MFAC App Developer's SDK.

MFAC comprises the following components:

Application Connectors: Available for web browsers, mobile applications, and custom application integration. The connectors provide a conduit for communication between web and mobile applications and the lower layers of the MFAC stack. In the case of web browsers, the connectors take the form of browser extensions in browsers that provide support for extensions.[5]

MFAC Service: Handles requests from the connector layer. The service orchestrates the message flow and overall operations of the client.

Figure 6
MFAC Architecture



5. In environments where browser extensions are not appropriate, non-extension-based interoperability layer designs are employed.

Authenticator Manager: Provides a unified way to access the functionality of the authenticators present on the device. The Authenticator Manager performs various authenticator management functions, such as discovering and tracking authenticators and their capabilities.

Authenticator Specific Module (ASM): Abstracts the details of an authenticator and the authentication process. ASMs perform the local authentication that unlocks the authentication key that is then used to authenticate the user to the MFAS server.

The authentication keys are stored in secure hardware within the authenticator when possible. ASMs also include the following component:

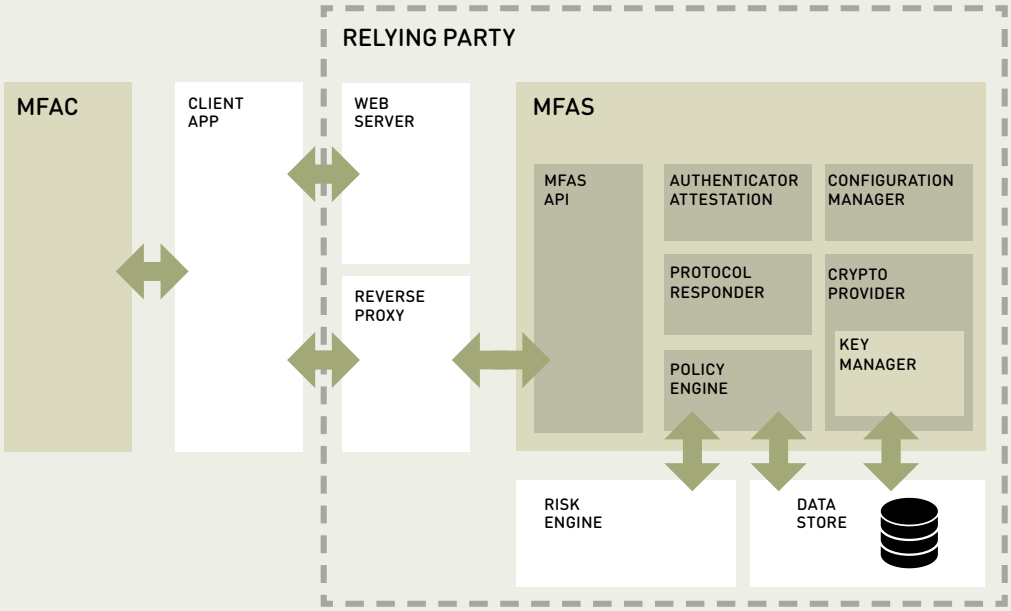
User Experience Layer: Responsible for presenting the authentication user experience flow during the authentication process. For example, a fingerprint ASM may display screens that guide the user to swipe her finger on a fingerprint sensor; a QR Code ASM may ask the user to take a picture of the computer screen with her mobile phone.

MULTIFACTOR AUTHENTICATION SERVER

Multifactor Authentication Server (MFAS) is an UAF server implementation that provides authentication services to relying party web and mobie applications. A simple integration with MFAS allows the relying party web application to plug into a standards-based, prebuilt authentication stack and use any supported authentication method available on users' devices (Figure 7).

MFAS authenticates clients by generating authentication challenges and validating the responses received from the client. The validation is performed using keys previously established with the client during the provisioning step. MFAS is also responsible for ensuring that the authenticator used by the client is genuine though an attestation process. To protect the user's privacy, no biometric information is transmitted to MFAS.

Figure 7
MFAS Architecture



MFAS consists of the following components:

MFAS API: Allows MFAC to communicate with the rest of MFAS. The communication happens over a secure TLS connection using a RESTful API. The MFAS API has been designed to require minimal effort for integration with the relying party application.

Protocol Responder: Receives UAF messages from the MFAS API component, translates them into internal commands, and triggers their processing. In the reverse direction, the Protocol Responder translates commands from other components into UAF messages and sends them to the client using the MFAS API component.

Crypto Provider: Implements the cryptographic algorithms and authentication protocols used by MFAS. The Crypto Provider also securely manages the keys used by the algorithms.

Authentication Attestation Manager: Responsible for validating the identity of the authenticator and the associated authenticator attestation keys before a user authentication is performed.

Policy Engine: Implements intelligence that determines which authenticator to use in a particular transaction, driven by policies configured by the MFAS administrator. Using the policy engine, organizations can perform risk-appropriate authentication that matches the authentication methods in use to the amount of risk involved in a transaction. For example, an online bank may configure a policy that allows customers to simply authenticate using face recognition when checking their account balance but requires additional fingerprint recognition to initiate an outbound transfer of funds.

Configuration Manager: Manages various configurations for MFAS, such as the type of authenticators accepted by the relying party. MFAS maintains a database of known authenticators and their security properties. Information stored includes the make and model of the authenticator, its ability to store keys, its ability to perform operations in secure hardware, and the associated attestation key. This information is used for multiple purposes:

— To validate the genuineness of the authenticator during the authenticator attestation process

— To determine known security properties of the authenticator

— To allow the Policy Manager and the relying party application to select the appropriate authenticator for a transaction, matching the authentication method to the risk involved in the transaction

SECURITY ADVANTAGES

Nok Nok Labs provides a high level of end-to-end security through numerous security features in the UAF protocol and the S3 Authentication Suite. An additional important contributor to security is the extensible and modular nature of the overall architecture. This extensibility makes it possible to smoothly migrate to even easier-to-use and more secure authenticators as they become available in the future, with no need to deploy additional systems.

PROTOCOL SECURITY FEATURES

The unique design of the UAF protocol provides protection against several types of attacks through these safeguards:

Minimal Use of Passwords: Instead of passwords, UAF relies on the use of strong cryptographic keys not visible to the user. This provides resilience against physical observation attacks such as shoulder surfing, filming the keyboard, recording keystrokes, and so on. Not exposing the key to the user also provides protection against phishing attacks (in which the user is tricked into disclosing the password). Using high entropy keys (128 bits or more) mitigates attacks based on guessing commonly used passwords.

No Transmission of Keys: Because UAF is a challenge-response based authentication protocol, the authentication keys never leave the device, providing resilience against eavesdropping attacks. Additionally, all communications between the client and the server are protected using a TLS connection.

Segmentation of Risk: UAF keys are unique on a per–user account, per-device, and per–relying party basis, providing risk segmentation. A compromise of any one of the three factors limits the breach to that factor only. For example, even if an attacker could recover keys from a stolen device with no secure hardware, other devices owned by the user would remain uncompromised. The relying party or the user could simply disable access from that device to plug the breach and prevent this one breach from spreading to other systems.

MFAC SECURITY FEATURES

Hardware Protection: MFAC is designed take advantage of a device's existing hardware security capabilities. Depending on the particular capabilities of the client platform, different levels of enhanced security implementations are possible:

— Authentication keys stored in hardware.

— Signing and verification implemented using the hardware capabilities of TPMs and secure elements.

— The MFAC Service layer, including the UAF implementation, execute in a trusted execution environment.

— The majority of MFAC executes in a trusted execution environment and can use secure keyboard input, secure display, and peripherals where available.

Software Protection: Independent of the availability of secure hardware on a user's device, MFAC uses several techniques to harden the client, including the following:

— Authentication credentials and confidential cryptographic keys are never stored in the clear.

— Executable code is subject to multiple code integrity checks.

MFAS SECURITY FEATURES

MFAS protects the system and the credentials stored on the server with a broad range of security safeguards:

— Cryptographic decoupling of server keys from client keys. If an attacker gains unauthorized access to user credentials stored on one MFAS, the attacker will not be able to impersonate a user on either the compromised server or any other MFAS server.

— Encrypted storage of keys in the MFAS database,[6] including server-side key storage integration with existing key management systems (KMS) and associated hardware security module (HSM) protection of user and system keys.

— Verification of the authenticator used by the client via authenticator attestation keys when supported by the specific authenticator.

— Attribution of a dynamic trust level to each authenticator by using risk scores from existing risk engines.

OUR PRODUCTS FIT WITHIN THE AUTHENTICATION ECOSYSTEM

The Nok Nok Labs solution addresses "first mile authentication" problems—that is, problems occurring between the user and the first relying party. First mile authentication is complementary to single sign-on (SSO) and federation solutions that allow the users to authenticate once and reuse the user's identity across multiple applications. Such solutions concentrate risk, as a breach at the point of authentication means multiple services are compromised. By interoperating with these solutions and allowing users to authenticate securely to them, Nok Nok Labs strengthens the overall security of the entire authentication chain.

MFAS can also be integrated with risk-based authentication (RBA) solutions to enhance their effectiveness. The client information available to MFAS—such as the user's device, authenticator make/model, and hardware key storage capabilities—can be combined with other information traditionally used by RBA systems to form a clearer picture of the user. The RBA system interoperates with the MFAS policy engine to determine which authentication method to accept in a particular transaction.

The extensible design of the Nok Nok Labs solution future-proofs organizations and enables them to support new devices and authenticators. New devices supporting the UAF standard will include clients preinstalled by OEMs during manufacture. New authenticators supporting UAF are accompanied by ASMs supplied by authenticator vendors and integrated by device OEMs. This allows relying parties to enable support for new authenticators by simply updating the MFAS known authenticators list and adding policy rules. Furthermore, in the event of an attack on a particular authenticator that breaches the integrity of an authenticator used by an organization, the organization can easily replace the authenticator with an alternative one present on the user's device. In this manner, Nok Nok Labs allows organizations to evolve with the changing authentication landscape.

6. Encrypted key storage, KMS, and HSM integration are supported, but not required.

CONCLUSION

The recent explosive growth of mobile devices and use of the cloud have drastically changed the way applications are designed and delivered. These changes render both password-based authentication and existing strong authentication methods inadequate to meet the needs of today's organizations and users. Organizations need to serve large-scale heterogeneous user populations characterized by wide diversity in the devices they use and the ways they use them. The lack of suitable authentication solutions has resulted in frustrated users, weakened security, and complex deployments. Nok Nok Labs addresses these problems by taking advantage of the existing security capabilities of devices to create a unified, extensible infrastructure that can provide authentication to any application on any device using any authentication method. With FIDO Ready compliance for UAF and U2F protocols, the solution works on the abstraction principle to enable interoperability between any application and any authentication method.

Nok Nok's Multifactor Authentication Client (MFAC) unleashes the security capabilities of billions of devices by plugging them into a standardized end-to-end framework. In a typical mode of operation, MFAC uses a local authentication to unlock an authentication key, which is then used to authenticate to the server.

The Multifactor Authentication Server (MFAS) integrates with replying party server applications and provides them with authentication services. MFAS validates the genuineness of authenticators on client devices using an attestation key. When authenticating users, MFAS generates challenge strings and validates responses from clients. MFAS allows organizations to set policies that match the risk involved in a transaction to the authentication method.

The Nok Nok Labs approach offers the following advantages:

— Unifies current authentication silos and reduces complexity in authentication implementations.

— Improves the user experience of authentication by minimizing password use and enabling user-friendly authentication methods such as face or voice biometrics.

— Provides more secure authentication by segmenting risk and taking advantage of secure hardware.

— Future-proofs organizations by allowing them to easily enable new authentication methods on new devices by simply configuring policies.

ABOUT NOK NOK LABS
Backed by a team of security industry veterans from PGP, Netscape, PayPal, and Phoenix, Nok Nok Labs has deep experience in building internet- scale security protocols and products. Our ambition is to fundamentally transform authentication, by unifying authentication into one standard protocol, giving business the power to make the utmost of the cloud, data, mobile, and business online.

Nok Nok Labs
4151 Middlefield Road, Suite 200
Palo Alto, CA 94303 USA

www.noknok.com

TO LEARN MORE ABOUT NOK NOK LABS, VISIT NOKNOK.COM OR CONTACT US AT INFO@NOKNOK.COM

Nok Nok
LABS