UNBOUND

UNBOUND CRYPTO-OF-THINGS FOR SECURING IDENTITY

Prof. Yehuda Lindell,
Chief Executive Officer and Co-founder

WHITEPAPER

Introduction

The challenge of verifying the identity of a human user is one of the most basic in computer security, with passwords being the classic solution used for decades. The death of passwords has been predicted for years; Bill Gates predicted their death in 2004, and he was far from the first to do so. However, they are still widely in use -- and while we may be able to significantly reduce the number of passwords that users have, their existence to some extent may always be here. The reason for this is that the use of something the user "knows" is in fact a powerful authentication mechanism. Unfortunately, a password alone is in practice very weak, since users choose bad passwords, forget them, reuse them in multiple accounts, and so on. As a result, where security is crucial (e.g., for bank accounts, access to enterprise systems in the cloud) the use of multiple authentication factors is preferable.

In general, an authentication "factor" is defined as something that you know (password), something you are (a physical trait such as fingerprint or face), or something you have (an additional device). No one factor itself is strong enough; passwords can be weak, devices can be stolen or breached, and biometrics can be bypassed. Thus, what is most commonly found is the combination of two factors – typically a device and either a password or biometric.

The most common existing solutions for the additional factor of something you have are:

1. Hardware OTP (one-time password) tokens: These devices generate one-time codes based on a cryptographic key stored inside the device. The same cryptographic key is also held by a server who can generate the same one-time password to verify that the value provided by the user is correct.

There are several variants of these types of tokens; the most common types are disconnected tokens that present a one-time password on a built-in screen. Some tokens have a keypad which can be used to achieve challenge/response functionality, or to require a user to enter a PIN code before a one-time password is displayed.

2. Standalone one-time password mobile applications: These provide the same functionality as hardware tokens but run in software. They support enrollment (usually via QR code) of multiple accounts. Both Microsoft and Google offer such free apps.

3. Soft token SDKs: This is software that can be embedded into mobile apps and utilizes cryptographic operations to authenticate the user and device. Such solutions can improve user experience because the user does not need to switch to a dedicated OTP app and there is no need to copy or type in one-time passwords. In addition, more advanced asymmetric cryptography can be used, like digital signatures, which have significant security advantages over one-time passwords.

4. SMS-based one-time passwords: This is a user-friendly method that does not require users to install any app. Rather, in order to authenticate, a one-time password is sent by SMS to the user's registered phone, and this is used to authenticate them.

5. Smartcards and cryptographic hardware tokens: Smartcards and cryptographic hardware tokens are physical devices that can perform cryptographic operations like decryption and signing, while providing strong physical protection of the keys inside a fully isolated secure enclave. They can be used for logon to PCs (e.g. via Windows Smartcard Logon) as well as to digitally sign transactions to verify that the authentic user indeed authorized this specific transaction. Smartcards require a dedicated reader or may be contactless; cryptographic hardware tokens are typically connected via USB.

The additional factors of a password or biometric can be added to any of the above methods by simply providing the additional authentication material separately. In the case of smartcards, passwords are typically integrated into the smartcard itself, requiring the user to authenticate via a password before carrying out any sensitive operations.

The above pre-existing solutions for the additional factor of "what you have" all face significant challenges that make them problematic in many settings.

In this white paper, we will describe these challenges and then present Unbound's novel CoT (crypto-of-things) solution for securing identity, which simultaneously achieves both high security and high usability.

The Problems with Existing Solutions

When considering different solutions, important parameters to consider are the security of the solution, the end-user experience (usability), and the ease with which the solutions are deployed and administered. We will analyze each of the five solution types described above with respect to these parameters.

Hardware one-time password tokens: Hardware one-time password tokens face many security challenges:

- Users can fall to social engineering attacks (where they are convinced to provide their one-time password to an attacker);
- Tokens are often left lying around, making them vulnerable; and
- Physical attacks can be used to extract the secret key from within.

Despite this, they still provide adequate security for basic authentication and thus have successfully prevented a large number of attacks.

It is worth noting, however, that hardware tokens are not a very good solution for more complicated authentication scenarios like when transaction signing is needed. In addition, they are vulnerable to server-side attacks since the same keys used for generating the one-time passwords must be stored on a server, making them vulnerable to mass theft.

The main problem with hardware tokens is that from the user experience side, they are a huge pain. Users need to have them with them whenever they need to authenticate and carrying them around everywhere is painful. For example, think about a situation where the user must authorize a bank transfer with a hardware token, but is away on vacation or even just out for dinner when the call comes in to complete an urgent transaction. In countries where hardware tokens are popular, users sometimes need to carry multiple tokens with them – one for each bank account, one for work, and so on. In addition, when one-time password tokens with keypads are used to bind the authentication to an actual transaction via challenge/response (an extremely important security feature), the user experience is even worse.

On the management and operations side, hardware gets lost or broken, and replacing tokens is both expensive and painful. To further compound this, since users who lose their tokens must be provided temporary access, this situation opens up a popular vector of attack, called a social engineering attack, where an attacker calls the help desk to impersonate the legitimate user saying that their device has been lost. Furthermore, procurement and deployment of hardware tokens are slow and costly, due to the inherent costs associated with physical hardware, and in general impose significant administrative burdens on organizations.

Software one-time password tokens: Software solutions, in the form of mobile apps for one-time passwords, solve many of the user experience and administrative pains of hardware – but not all. In some cases, the user experience is still not great due to the need to switch between different mobile apps (e.g., if the one-time password is needed for authenticating within a mobile app).

In addition, since the organization does not control the 3rd party app, its functionality is very limited, making it challenging to support users. When authenticating a mobile app, many of these challenges can be addressed by an organization building such functionality directly into the app. However, deploying cryptographic solutions correctly is known to be extremely difficult, and far more complex than initially estimated. Even if one were to do so, the primary problem with all such solutions – whether they are 3rd party authentication apps or integrated software – is due to inherent vulnerabilities of mobile phones.

A maliciously designed app – one with great functionality and provided for free – can be used to steal the cryptographic keys in software tokens on mobile phones, completely breaking security. To make things even worse, once stolen, the attacker can generate one-time passwords in parallel to the legitimate user and go completely undetected.

Soft token SDKs: These solutions have almost the same properties as software one-time password tokens, except that asymmetric cryptography can be used for digital signing. In addition, usability is better since the user does not need to switch between apps and enter one-time passwords. However, the security vulnerabilities are the same, and therefore extremely problematic when a good level of security is needed.

SMS-based one-time passwords: In general, SMS-based one-time password authentication is user-friendly. However, problems can arise when the SMS does not arrive due to a carrier issue. Since the user has no way of determining where the problem lies, costly help desk calls to the organization are often the result. In addition, SMS often does not work when the user is overseas, resulting in an inability to authenticate.

Beyond these issues that can arise, the primary problem with SMS-based authentication is security. Many attacks based on SIM swapping, SS7, and malware, have proven very effective. For example, some attackers use identifying information about the victim to convince the cellphone provider that they have changed their phone. These security risks resulted in NIST recommending that SMS-based one-time passwords not be used any more.

Smartcards: Smartcards suffer from the same usability, management, and administration pains as hardware one-time password tokens. They have the security advantage of being able to sign on actual transactions – binding the identity to the transaction itself – as well as being far less vulnerable to social engineering. However, the challenges that arise from an operations point of view are so great that they are not used nearly as much as they should be.

A New Paradigm for Software-Defined Cryptography

As we have shown, hardware-based solutions provide high security – but also suffer from low usability, high expense and challenging administration and management. In contrast, software solutions are the mirror of hardware solutions: they have good user experience, and are much easier to deploy and manage, but offer poor security. Ultimately, enterprises and end-users need a software solution that combines both good user experience and high security.

Secure Multiparty Computation for Software-Defined Cryptography

To achieve a software-based solution that is also secure, we need a different security model. In particular, the cryptographic key or credential cannot be stored on a mobile device (or laptop) and cannot reside in memory of any single machine at any given time. Otherwise, as we have discussed, it is vulnerable to being stolen.

This may appear to be an impossible task – how can one carry out cryptographic operations such as one-time password generation, decryption, or transaction signing, without holding the key? Fortunately, a technology called secure *Multi-Party Computation (MPC)* – also known as *threshold cryptography* in the context of protecting keys – does just that.[1] MPC works by splitting the secret key into two parts called shares, such that no individual share reveals any information about the key. Then, one share is placed on the mobile phone (or another endpoint device) and the other share is placed on a server. MPC protocols enable the mobile and server to interact with each other in order to obtain the result of the cryptographic operation (e.g., digital signature or one-time password) without the mobile or server revealing to each other their own share or any other information about the key. This

[1] MPC has been studied in academia for over three decades and has a strong and well-founded theory. MPC protocols have mathematical proofs of security, guaranteeing that an attacker who is unable to breach both devices is unable to learn anything whatsoever about the key, even if the attacker knows the protocols being used and can run arbitrary malicious code in its attempt. More basic information about MPC can be found in a Basic introduction to MPC and a Technical Primer to MPC. A more in-depth introduction to MPC for a general computer science audience appears here, and resources for in-depth study of MPC can be found at the MPC alliance page and here.

means that the key remains fully protected, even while in use. In particular, the key cannot be stolen from the mobile since it never appears there in complete form. Likewise, users' keys cannot be stolen from the server if it is breached, since the key does not appear there either; this also means that the server cannot carry out the operation without the user's approval.

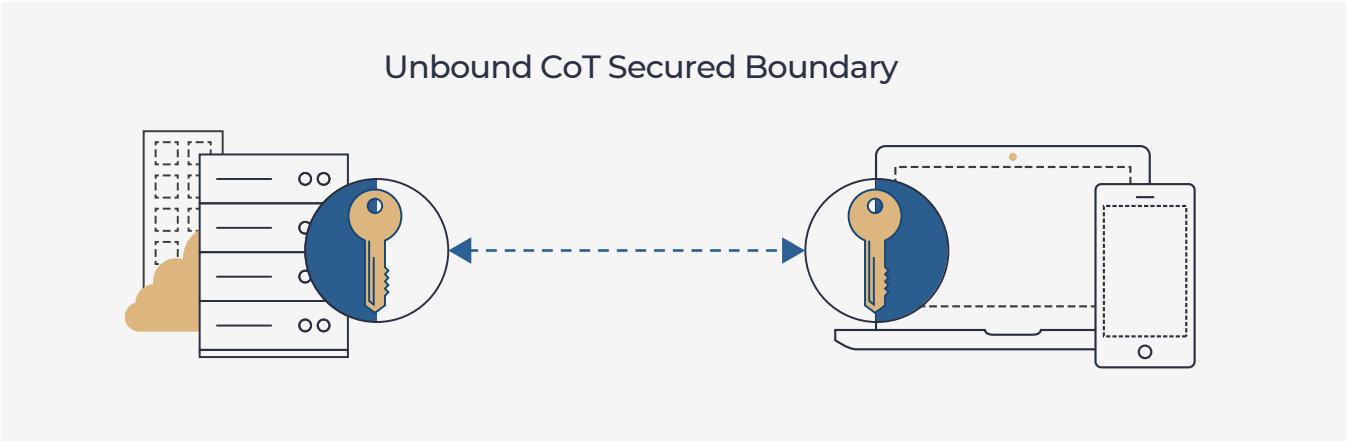
Unbound's Academic and Research Background

Unbound was founded by MPC researchers. Both Professor Nigel Smart and myself have been researching MPC for many years, and have collectively written approximately 100 papers on the topic (see Nigel's publication list and my publication list). Several of Unbound's MPC protocols are the fruit of internal research at the company (for just one example, the two-party protocol for ECDSA published at CRYPTO 2017). Unbound's team of cryptographers ensure that the MPC solutions that we provide undergo rigorous inspection and evaluation internally before deployment.

As cryptographers and researchers, we strongly believe in transparency and independent review. All novel protocols used by Unbound have been published in open academic conferences and have been peer reviewed. In addition, an independent review of all of Unbound's protocols was undertaken by Professor Victor Shoup of NYU, and a separate independent code review that the actual code matches the specification reviewed by Prof. Shoup was also carried out. Finally, as part of our belief in transparency, all information about our protocols (and even the code) is available to customers, under NDA.

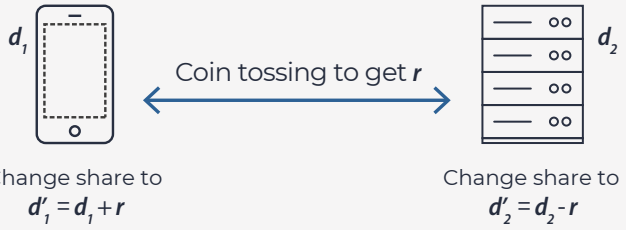
Unbound's CoT Solution

Unbound's crypto-of-things (CoT) solution uses MPC to split keys between a server and a mobile device (or other endpoint, like a laptop), and to carry out computations without ever uniting the key shares. Unbound's CoT is integrated into existing mobile or desktop apps using standard cryptographic libraries. Every time an operation is called, an MPC protocol is triggered between the mobile and server, using the key shares held by the mobile and the server to complete the operation.



In addition to splitting each key into shares (parts) that are never united, the key shares are refreshed with every single operation, so that although the key remains the same, the material held by each device is completely re-randomized. This means that a share stolen from a mobile device by an attacker will become completely useless after a refresh. See the diagram below for an example of how this works.

- Initial key sharing of d: random values d_1 and d_2 so that $d_1 + d_2 = d$
- The refresh process:



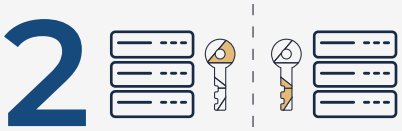
- Note that $d'_1 + d'_2 = d$ and so is a valid sharing
- Note that given d_1 and $d'_2 = d_2 - r$, nothing can be learned about d

Note that even if an attacker manages to completely clone the device (a difficult task) and run an MPC operation before the legitimate mobile device, then the refresh on the server side will render the share held by the legitimate mobile as invalid. This guarantees detection of any fraudulent operation, which is an extremely important property.

The security model offered by Unbound's CoT solution is completely different to that of hardware. Instead of holding keys in a single place and using physical and software hardening to prevent access, the keys are distributed between two locations so that access to both is needed to learn anything.

Given the inherent strong separation between devices held by users and servers in the cloud or in an organization's data center, it is extremely difficult for an attacker to gain access to both shares of the key. This is further compounded due to the refresh procedure, which guarantees that an attacker needs to access both the user's device and server at essentially at the same time. This provides a very strong security guarantee without compromising on the functionality, flexibility, and benefits of software.

Each private key exists as two separate random shares stored on separate locations & refreshed constantly



Key shares never combined at any point in time - not even when used or when created

Key material never exists in the clear at any point of its lifecicle

Loyal to the defense-in-depth approach, various software and hardware security technologies are used in synergy and not considered mutually exclusive. For example, when the mobile has a secure enclave that can carry out signing, the MPC protocol messages are signed using a key in the enclave, providing even stronger device binding. Unbound's CoT uses whatever local security measures are available, while providing a unified interface and high level of security for all types of devices.

Due to the rich capabilities of MPC, Unbound's CoT supports all standard cryptographic operations and algorithms: RSA decryption and signing, one-time password generation, Elliptic curve cryptography, AES, HMAC, and so on. With CoT, the mobile is no longer limited to only generating one-time passwords and can be used to sign on transactions and carry out any cryptographic operation, from anywhere.

Unbound's CoT achieves the notion of software-defined cryptography, in the sense that it constitutes a solution that previously required hardware to be securely deployed. This transition to a software-only solution means that users' mobiles can be safely used for cryptographic operations, and usability no longer needs to be at odds with high security. As we will see in the next section, this provides numerous benefits.

Features and Benefits of Unbound's CoT Solution

Unbound's CoT is a type of soft token SDK, that differs from other software solutions primarily in the fact that by utilizing MPC it achieves a high level of security. As such, Unbound's CoT is a library that can be integrated into any app, and provides strong protection of any cryptographic key used by the app.

In a simple scenario, CoT can be used to build an app that generates one-time passwords on a mobile, like other existing apps, but while providing high security of the cryptographic key used to generate the one-time passwords. However, it can actually do much more:

1. CoT can be integrated into any mobile app so that the additional factor of authentication is fully transparent to the user.

2. It can be used in conjunction with web authentication -- e.g., when carrying out operations on the web and not just in a mobile app -- so that the user's mobile behaves as a separate out-of-band authenticator.

3. It can be used for any device, including laptops, PCs, and so on.

In all of these cases, it is possible to not only generate one-time passwords, but also digital signatures or any other standard cryptographic operation.

In the remainder of this section, we analyze Unbound's CoT under four parameters: security, user experience, ease of deployment, and ease of administration.

Security

As we have described above, Unbound's CoT solves the single point of security failure inherent in legacy solutions. The key is never whole in any place at any time, and the key is fully protected even in the case of a complete breach of a mobile or of the CoT server. Furthermore, due to the refresh procedure, an attacker needs to essentially breach both (completely different) environments and break all of the local protections before any refresh takes place.

Beyond the core MPC solution, Unbound's CoT uses any available local secure enclave or trusted execution environment to further bind the solution to the device, and all operations are logged both on the mobile and on the server. This provides extremely strong device binding, together with high visibility and management capabilities.

In addition, since Unbound's CoT provides the ability to execute cryptographic operations, transactions can be digitally signed using standard cryptographic signing keys and based on the "what-you-see-is-what-you-sign" approach. This has the advantage of strong binding to the transaction and non-repudiation. In addition, the authentication server only needs to store public keys, simplifying the protection mechanisms required for this server. In a scenario where one-time passwords are used (e.g., in order to make transition easy by not changing existing systems), CoT can be used in conjunction with Unbound's vHSM solution to protect the seeds used to calculate the one-time passwords on the backend authentication server.

The transition from one-time passwords to the use of asymmetric digital signatures (e.g., RSA and ECDSA) with Unbound removes two significant vulnerabilities of classic one-time password solutions:

1. One-time passwords are based on shared (symmetric) secrets that are vulnerable on both the client (mobile phone) and server side. By moving to digital signatures, the server only holds public keys. Thus, the threat of mass theft of users' credentials from the server is completely removed.

2. Second, the use of MPC enables the private key used for digital signatures to be strongly protected, even in software. This mitigates the threat of theft of users' credentials from breached clients.

Another significant security benefit is due to the fact that since Unbound's CoT is integrated into the mobile app, there is no separate one-time password that is presented to users. This reduces the possibility of social engineering attacks where users are tricked into giving one-time passwords to attackers, since either digital signatures are used (and so there are no one-time passwords) or one-time passwords are sent without the users seeing them and thus without the ability to give them to anyone else.

Historically, the use of a mobile phone as a second factor of authentication has been considered weak, due to the low level of security on these devices. However, once this problem has been dealt with, mobile phones actually have security advantages over one-time password tokens and smartcards. First, mobile phones are personal devices that people usually do not leave lying around, unlike physical tokens that can then be briefly accessed by an attacker. Second, if a physical token is stolen, it can take a while until this is noticed by the legitimate user. In contrast, if a user's mobile phone is stolen, this will be detected quickly, and the ability to use the authentication on that phone can be immediately revoked.

We conclude by remarking that great flexibility can be achieved using Unbound's CoT, enabling the construction of sophisticated security mechanisms. For example, consider the use of a user's mobile in conjunction with a PC to provide a separate out-of-band authentication mechanism. Assume that the user is accessing a sensitive system in an organization or carrying out a large financial transaction from their PC. In such a case, the access or transaction information can be pushed to the user's mobile for authorization and digital signing using Unbound's CoT. This significantly mitigates man-in-the-browser attacks, since the access/transaction information can be presented to the user on their mobile, which is a completely different device, before they approve.

User experience

The user experience with an integrated strong authentication solution is optimal. It is completely transparent to the user, who authenticates using a biometric, password, swipe, or whatever is desired by the organization. The strong device binding and security provided by Unbound's CoT is completely invisible to the user. In cases where organizations wish to continue using existing one-time passwords in order to be compatible with existing systems, the user experience can be exactly the same as with software one-time password apps. It is also possible to have the MPC calculated OTP auto-sent for authentication. The fact that a much higher level of security is achieved has no impact whatsoever on the user's experience.

Ease of deployment

Unbound's CoT solution is pure software and so there are no issues with procuring and delivering physical hardware. This is especially important when users are remote or traveling, and securely delivering hardware tokens is expensive and slow. Furthermore, when integrating the solution into an existing mobile app, the deployment is fully streamlined. This removes the requirement of installing, configuring, and maintaining two separate apps in order to do business with strong authentication.

Another important property is that existing systems can be easily transitioned with minimal changes. For example, in organizations where the authentication server only accepts standard passwords, it is possible to use MPC between the mobile and CoT server to deliver the user's password to the authentication server without the user typing it in and without the password being vulnerable. This means that security can be immediately boosted by making changes only on the mobile side, and without modifying the authentication server (which is often run by a different team). At a later stage, the authentication server can be upgraded to accept digital signatures or other forms of strong authentication, if desired.

Ease of administration

The administration of hardware tokens and smartcards is extremely painful, and the fact that Unbound's CoT is a software-only solution immediately makes administration much easier. However, it goes well beyond that. When using separate one-time password authentication, the need to administer two separate solutions is a significant headache. By integrating Unbound's CoT, the administration overhead is identical to that of a single app, as if no strong authentication is used at all.

Other Use Cases

Our above description and examples have focused on the use of a mobile phone as a second factor of authentication, a popular use-case for Unbound's CoT solution. However, it is important to stress that Unbound's CoT is also relevant for many other use cases where devices require secure identity and operations.

The following list contains just a few examples of relevant use cases:

1. Replacing smartcards on PCs (for smartcard logon, transaction signing, and more).
2. Securing access to cloud environments via strong cryptographic signing from PCs and mobile phones.
3. Security IoT devices, where cryptographic keys are essential for authenticating the device, but secure hardware is typically not an option.
4. Securing cryptocurrency wallet seeds that protect significant funds and are therefore an attractive target.
5. Replacing set-top-box smartcards, where the strong device binding prevents cloning and other attacks.

Summary

In summary, Unbound's CoT solution enables organizations to break away from the classic usability/security dilemma that forces them to either accept weak security or bad user experience and painful administration. By using advanced MPC techniques to secure cryptographic keys on any device, anywhere, it is possible to achieve the benefits of software solutions without compromising on security. The result is the ultimate combination of security, user experience, operations, and cost effectiveness.

Type	Security	End-User Experience	Ease of Deployment	Ease of Administration	Cost
Hardware OTP	Medium-High	Difficult	Difficult	Difficult	High
Software OTP	Low	Moderate-Easy	Easy	Moderate-Easy	Low
Soft-token SDK	Low	Easy	Easy	Moderate-Easy	Low
SMS OTP	Low	Easy	Easy	Easy	Medium
Smartcards	High	Difficult	Difficult	Difficult	High
Unbound	High	Easy	Easy	Moderate-Easy	Low