دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۰، جلد ۱۹، شماره ۱ (۳۴-۴۲)
https://monadi.isc.org.ir

ارائه‌شده در شانزدهمین کنفرانس بین المللی انجمن رمز ایران در دانشگاه فردوسی مشهد، ۶ و ۷ شهریور ۱۳۹۸

# ارائه مدل عملی حفظ حریم خصوصی در قراردادهای هوشمند مبتنی بر زنجیره بلوکی با کاهش سربار★

امیر میرزایی*، سید محمدحسین فرزام و سیاوش بیات سرمدی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

**اطلاعات مقاله**

کلمات کلیدی:
رمزارز
زنجیره بلوکی
قراردادهای هوشمند
مزایده دومین قیمت
حریم خصوصی
اثبات صفر دانش

dor: 10.0000/000000000

نوع مقاله: پژوهشی

**چکیده**

در سال‌های اخیر با رشد روزافزون رمزارزها و فناوری زنجیره بلوکی، مدل‌های مختلفی از ارزهای دیجیتال ارائه شده است. هر یک از این رمزارزها دارای ویژگی‌های خاص خود می‌باشند. رمزارز اتریوم دارای قابلیت نوشتن قطعه کدهایی در داخل بلوک‌های زنجیره بلوکی می‌باشد که به صورت خودکار اجرا می‌شوند و به قراردادهای هوشمند معروف هستند. در اتریوم این قطعه کدها به صورت شفاف و بدون حفظ حریم خصوصی قابل اجرا هستند؛ این در حالی است که محرمانگی و حفظ حریم خصوصی از مهم‌ترین مولفه‌های امنیت در حوزه داده و شبکه می‌باشد. پیش‌تر نمونه‌هایی مانند اِنیگما، شَدوات و هاک برای رسیدن به این مهم ارائه شده است که دارای سربار قابل توجهی می‌باشند. همچنین بخشی از این نمونه‌ها خارج از زنجیره اجرا می‌شوند که دارای معایب خاص خود هستند. در این مقاله با بهبود مدل هاک ساز و کاری برای حفظ حریم خصوصی در قراردادهای هوشمند ارائه شده است که سربار محاسباتی زمان اجرا را به میزان قابل توجهی کاهش می‌دهد. نتایج پیاده‌سازی مزایده دومین قیمت با استفاده از راهکار پیشنهادی نشان از بهبود نزدیک به ۵۰ درصدی در زمان اجرای قرارداد هوشمند در سمت مدیر دارد.



## ۱ مقدمه

در چند سال اخیر ارزهای دیجیتالی مختلفی با عنوان رمزارز[1] ارائه شده است که اساس کار آن‌ها فناوری زنجیره بلوکی[2] و رمزنگاری[3] می‌باشد. این فناوری به کاربران اجازه می‌دهد تا در یک شبکه عمومی، توزیع شده[4] و غیر قابل اعتماد اقدام به تبادل ارزهای خود کنند. چهار مولفه امنیت یعنی انکارناپذیری[5]، دسترسی‌پذیری[6]، صحت داده[7] و احراز هویت[8] در اکثر رمزارزهای معرفی شده وجود دارد [۱]. در مقابل، محرمانگی[9] و حریم خصوصی[10] ویژگی‌هایی هستند که در تضاد با فناوری زنجیره بلوکی قرار دارند و تامین آنها سربار زیادی به شبکه تحمیل می‌کند [۲].

پس از رمزارز بیت‌کوین[11] [۳] که تراکنش[12] مالی شفاف و بدون حریم خصوصی را برای افراد در شبکه فراهم کرد، رمزارز زی‌کَش[13] [۴] ارائه گردید. این رمزارز شامل هسته بیت‌کوین می‌باشد و امکان انجام تراکنش‌های مالی خصوصی را برای کاربران فراهم می‌کند. شایان ذکر است که رمزارز بیت‌کوین سطحی از ناشناس بودن[14] را برای کاربران ایجاد می‌کند اما رمزگذاری نشدن تراکنش‌ها باعث می‌شود امکان نقض حریم خصوصی کاربران با تحلیل تراکنش‌ها وجود داشته باشد [۳]. در بیت‌کوین و زی‌کش امکان اجرای قرارداد هوشمند[15] وجود ندارد و به همین دلیل

★از کمیته علمی شانزدهمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.

*نویسنده مسئول
آدرس‌های رایانامه: amircris@ce.sharif.edu (امیر میرزایی)، mfarzam@ce.sharif.edu (سید محمدحسین فرزام)، jalili@sharif.edu (سیاوش بیات سرمدی)



[1] cryptocurrency [2] blockchain [3] cryptography [4] distributed [5] non-repudiation [6] availability [7] data integrity [8] authentication

[9] confidentiality [10] privacy [11] Bitcoin [12] transaction [13] Zcash [14] anonymity [15] smart contract

گنجانده شدن این ویژگی در رمزارز اتریوم [1] [۵] باعث رشد سریع و فراگیر شدن این رمزارز در بین عموم مردم گردید. قراردادهای هوشمند قابلیتی می‌باشد که در آن کاربران می‌توانند قطعه کدهای تورینگ-کامل [2] را بر روی زنجیره بلوکی اجرا کنند [۵]. نتیجه این قطعه کدها در بلوک‌ها ثبت و نگهداری می‌شوند. در این سامانه، تمامی اعضا به گزارش‌ها و داده‌های قراردادهای هوشمند دسترسی دارند. یک تراکنش تنها در صورتی در زنجیره بلوکی ثبت می‌گردد که مورد توافق اکثریت اعضای شبکه قرار گرفته باشد. این نحوه انتشار اطلاعات و وجود ساز و کاری برای اجماع [3] باعث ایجاد اطمینان از صحت داده‌ها و دسترسی‌پذیری می‌شود [۶]، اما مشکلاتی همچون نقض محرمانگی و حریم خصوصی را به دنبال دارد. به عنوان نمونه، در مزایده دومین قیمت [4] که فرد برنده دومین مبلغ بالای پیشنهادی را پرداخت می‌کند، قیمت‌های پیشنهادی اطلاعات بسیار حساسی هستند که باید در طول مزایده محرمانه بمانند [۷].

محققان در سال‌های اخیر تلاش‌های وسیعی برای حل این مشکل انجام داده‌اند [۸]. مدل‌های حافظ حریم خصوصی مختلفی برای قراردادهای هوشمند ارائه شده است که هر یک دارای مشکلات خاص خود هستند. یکی از اولین مدل‌های معرفی شده سکوی [5] محاسباتی غیرمتمرکز اِنیگما [6] [۹] می‌باشد. این مدل یک شبکه نظیر به نظیر [7] است که افراد مختلف را برای ذخیره و اجرای محاسبات به طور مشترک به کار می‌گیرد به طوری که داده‌های قراردادهای هوشمند کاملا محرمانه بمانند. مدل محاسباتی انیگما براساس نسخه بهینه شده محاسبات چندجانبه امن [8] شکل گرفته است و توسط یک طرح تسهیم راز [9] و قابل اعتبارسنجی [10] تضمین شده است. داده‌ها بین گره‌های مختلف تقسیم شده و بدون اینکه اطلاعاتی به سایر گره‌ها فاش شود، توابع محاسباتی انجام می‌شوند. هیچ گره‌ای به تنهایی به کل داده‌ها دسترسی ندارد و هر یک فقط بخشی بی‌معنا از آن را در اختیار دارد. انیگما برای اتصال به زنجیره بلوکی موجود و بارگذاری محاسبات خصوصی بر روی شبکه، خارج از زنجیره طراحی شده است. این سکو وظیفه اجرای محاسباتی که نیاز به حریم خصوصی دارند را برعهده دارد. اثبات درستی اجرای محاسبات بر روی زنجیره بلوکی ذخیره می‌شود و می‌تواند توسط همه اعضای شبکه بررسی شود. سربار محاسباتی برای تخصیص و مدیریت داده‌ها در محاسبات چندجانبه امن هزینه فراهم نمودن حریم خصوصی برای داده‌های قراردادهای هوشمند می‌باشد.

مدل دیگری که در سال‌های اخیر ارائه شده است سامانه شَدوات [11] [۱۰] می‌باشد که با استفاده از سخت‌افزار اضافی، محرمانگی را برای قراردادهای هوشمند فراهم نموده است. این مدل یک چارچوب محافظت شده توسط محیط اجرایی قابل اعتماد [12]، خارج از زنجیره بلوکی عمومی مقرر کرده است تا قراردادهای هوشمند خصوصی در آن اجرا و ذخیره شوند. علاوه بر این در خارج از زنجیره، از سامانه ذخیره‌سازی توزیع‌شده محیط اجرایی مورد اعتماد [13] برای ذخیره کد دودویی، حالت‌ها و وضعیت قراردادهای هوشمند استفاده شده است. این سامانه توسط سخت‌افزار امن محافظت می‌شود و تغییر و افشای داده‌های آن امکان‌پذیر نمی‌باشد. نیاز به سخت‌افزارهای اضافی، تحمیل سربار محاسباتی فراوان برای تخصیص کارها و تولید اثباتِ درستیِ اجرا از معایب این مدل هستند. علاوه بر این موارد، در هر دو مدل گفته شده بخشی از کارها در خارج از زنجیره انجام می‌شود که در نتیجه دسترسی پذیری نتایج حاصل از این کارها کاهش پیدا می‌کند. شایان ذکر است که علاوه بر انیگما و شَدوات مدل‌های دیگری همچون اِکیدن [14] [۱۱] نیز وجود دارند که مشابه این اشکالات در آن‌ها نیز وجود دارد.

یکی از بهترین مدل‌های ارائه شده برای حل مشکل حریم خصوصی قراردادهای هوشمند، مدل هاک [15] [۱۲] می‌باشد. در این مدل سعی شده است با استفاده از یک شخص سوم و اثبات صفر دانش [16] بدون اینکه داده‌های قراردادهای هوشمند در دسترس افراد غیرمجاز قرار گیرد آن‌ها را اجرا کنند. در این سامانه قراردادهای هوشمند شامل دو قسمت خصوصی و عمومی می‌باشند. قسمت خصوصی قراردادهای هوشمند توسط شخص سوم اجرا می‌شود که به آن مدیر می‌گویند. به دلیل استفاده از اثبات صفر دانش در این مدل، لزومی ندارد مدیر قابل اعتماد باشد و هر یک از گره‌های موجود در شبکه می‌توانند به عنوان مدیر برگزیده شوند. با این حال پس از اجرای قرارداد هوشمند مدیر می‌تواند داده‌های قرارداد هوشمند را افشا کند. قسمت عمومی قراردادهای هوشمند به تعیین جریمه برای کاربران متخلف در روند اجرای قرارداد هوشمند مربوط می‌شود. به عنوان نمونه در صورت افشای داده‌ها توسط مدیر پس از اجرای قرارداد هوشمند، این کاربر باید جریمه‌ای به طرفین قرارداد پرداخت کند. در این مدل سربار تولید اثبات درستی اجرای قرارداد هوشمند توسط مدیر و تولید اثبات‌های دیگری در دیگر مراحل باعث افزایش زمان اجرای قراردادهای هوشمند شده است. همچنین پیاده‌سازی این مدل با توجه به محدودیت‌های موجود در اثبات صفر دانش و زیکَش با سختی زیادی مواجه بوده است.

در این مقاله با شخصی‌سازی مدل هاک برای مزایده دومین قیمت، کارایی آن بهبود داده شده است. در پیاده‌سازی انجام شده از روش پیشنهادی، رمزارز زیکَش به عنوان بستر پایه انتخاب شده است. نتایج ارزیابی حاصل از اجرای این قرارداد هوشمند بر روی رایانه‌ای با پردازنده core-i7 نشان از بهبود نزدیک به ۵۰ درصد در زمان اجرای قرارداد هوشمند در سمت مدیر دارد.

در ادامه این مقاله و در بخش ۲ به بررسی مفاهیم اولیه مورد نیاز پرداخته شده و در بخش ۳، پروتکل شخصی سازی شده پیشنهادی برای حفظ حریم خصوصی در قراردادهای هوشمند ارائه می‌گردد. بخش ۴ به نتایج ارزیابی پرداخته و در نهایت بخش ۵ به نتیجه‌گیری مباحث این نوشتار می‌پردازد.

[1] Ethereum [2] Turing-complete [3] consensus [4] second-price auction [5] platform
[6] enigma [7] peer to peer [8] secure multi-party computation [9] secret sharing
[10] verifiable [11] ShadowEth [12] Trusted Execution Environment (TEE)
[13] TEE Distributed Storage (TEE-DS) [14] Ekiden [15] Hawk [16] zero-knowledge proof

## ۲ مفاهیم اولیه

در این بخش مفاهیم مورد نیاز جهت ارائه پروتکل شخصی‌سازی شده برای حفظ حریم خصوصی قراردادهای هوشمند توضیح داده می‌شود.

### ۱.۲ زنجیره بلوکی

زنجیره بلوکی یک دفترکل[1] توزیع‌شده[2] غیرمتمرکز و بدون نیاز به اعتمادسازی است که توسط همه اعضا نگهداری می‌شود. بعضی از اعضا به عنوان معدن‌کاو[3] در این شبکه نظیر به نظیر فعالیت می‌کنند و یک نسخه کامل از زنجیره بلوکی را نگه می‌دارند. این اعضا تمام تراکنش‌هایی که توسط کاربران امضا و ارسال شده است را جمع‌آوری می‌کنند و پس از اعتبارسنجی امضاها، آن‌ها را در داخل یک بلوک قرار می‌دهند. هر بلوک اطلاعاتی از تراکنش‌ها را به همراه چکیده[4] بلوک قبلی نگه می‌دارد و لیست ترتیبی از این بلوک‌ها یک زنجیره بلوکی را تشکیل می‌دهند. زمانی که یک بلوک توسط یک معدن‌کاو تولید می‌شود، تمام معدن‌کاوان موجود در شبکه باید برای قبول کردن و اضافه کردن آن به زنجیره به اجماع کلی برسند [۱]. فناوری زنجیره بلوکی کاربردهای زیادی در بخش‌های مختلف دارد. از این فناوری می‌توان در حفظ امنیت شبکه اینترنت اشیا [۱۳] و انجام تراکنش‌های مالی امن در صنایع مختلف به ویژه سازمان‌های بیمه و بازرگانی استفاده نمود.

### ۲.۲ زی‌کَش

زی‌کَش نسخه‌ای از یک طرح پرداخت ناشناس غیرمتمرکز به نام زیروکش[5] است که از نظر امنیتی و کارایی بهبود یافته است [۴]. این رمزارز با ترکیب رمزارز بیت‌کوین برای تراکنش‌های مالی و زی‌کی–اسنارک[6] برای محافظت از این تبادلات ساخته شده است. برای حفاظت از تراکنش‌های مالی در یک رمزارز، باید ساز و کاری وجود داشته باشد که علاوه بر جلوگیری از دسترسی غیرمجاز به اطلاعات حساب، رفتار و عادت‌های مالی کاربران امکان تصدیق اعتبار تراکنش‌ها را نیز زیر سوال نبرد. در این طرح‌ها از اثبات صفر دانش برای محافظت در برابر تجزیه و تحلیل نمودار تراکنش‌ها استفاده شده است.

مشابه بیت‌کوین در زی‌کش نیز حساب کاربری وجود ندارد و دارایی‌های هر کاربر همان تراکنش‌های خرج‌نشده هستند. در زی‌کَش دو نوع تراکنش شفاف و محافظت شده وجود دارد. چارچوب تراکنش‌های مالی شفاف همانند بیت‌کوین است و تراکنش‌های محافظت شده توسط یادداشت‌ها[7] مشخص می‌شوند [۱۴]. در این رمزارز هر کاربر می‌تواند آدرس‌های خصوصی متعددی داشته باشد و هر یادداشت می‌تواند متعلق به یکی از این آدرس‌ها باشد. برای هر یادداشت تعهد[8] متناظری وجود دارد. در این رمزارز برای تولید تعهد از تابع چکیده‌ساز SHA256 مطابق رابطه (۱) استفاده می‌شود [۱۴]:

$$CM := \text{SHA256}(a_{pk}, value, rho, s). \quad (۱)$$

در این رابطه $a_{pk}$ کلید عمومی صاحب یادداشت، $value$ مقدار پولی که یادداشت مورد نظر دارد و $s$ تک‌شمار[9] تصادفی برای تعهد می‌باشد که در اصطلاح به آن دریچه[10] گفته می‌شود. در صورتی که یادداشت مورد نظر حاصل تراکنش‌های مالی خصوصی باشد مقدار $rho$ طبق رابطه (۲) با استفاده از تابع چکیده‌ساز تولید شده و در غیر اینصورت به صورت تصادفی تعیین می‌گردد [۱۴]:

$$rho := \text{SHA256}(phi, h_{sig}). \quad (۲)$$

در این رابطه $phi$ یک عدد تصادفی و $h_{sig}$ مقداری وابسته به باطل‌کننده‌های تعهدهای قبلی این تراکنش است. باطل‌کننده[11] مقدار منحصر به‌فردی در زنجیره بلوکی است که مرتبط با تعهد نظیرش می‌باشد. صادر نمودن باطل‌کننده برای هر تعهدی در زنجیره بلوکی باعث می‌شود تعهد مورد نظر دیگر اعتباری نداشته باشد. این کار زمانی اتفاق می‌افتد که صاحب تعهد به دنبال خرج کردن پول خود باشد. این مقدار فقط توسط صاحب تعهد می‌تواند صادر شود که دارای کلید $a_{sk}$ می‌باشد. این کلید همان کلید خصوصی کاربر است [۱۴]. یک باطل‌کننده مطابق رابطه (۳) تولید می‌شود:

$$Nullifier := \text{SHA256}(rho, a_{sk}). \quad (۳)$$

در زی‌کَش تراکنش‌ها در قالب مفهومی به نام جوین‌اسپلیت[12] انجام می‌شود که شامل تبادلات مالی شفاف و محافظت شده می‌باشد. هر تراکنش مالی محافظت شده شامل ابطال‌کننده‌های تعهدهای خرج شده، تعهدهای جدید تولید شده، ریشه درخت مرکل و اثبات درستی این تراکنش می‌باشد. درخت مرکل داده‌ساختاری است که نگه‌دارنده تمام تعهدهای موجود در زنجیره بلوکی است. هر تراکنش جوین‌اسپلیت دارای یک جفت کلید خصوصی و عمومی می‌باشد که در تولید $h_{sig}$ و اثبات درستی استفاده می‌شود. برای تولید اثبات موجود در جوین‌اسپلیت از زی‌کی–اسنارک استفاده می‌شود [۱۵]. این اثبات به کاربرانی که قصد اعتبارسنجی ابطال‌کننده‌ها و تعهدهای جدید را دارند اجازه می‌دهد تا بدون فاش شدن ورودی‌های رابطه‌های (۱)، (۲) و (۳) از درستی نتیجه آنها اطمینان حاصل فرمایند. در نهایت این کار باعث می‌شود کاربران موجود در شبکه برای ثبت این تراکنش مالی در بلوک‌های زنجیره بلوکی به اجماع برسند.

### ۳.۲ زی‌کی–اسنارک

در زی‌کَش برای اثبات صفر دانش از زی‌کی–اسنارک استفاده شده است که معرف اثبات صفر دانش با اندازه پیام کوتاه و بدون نیاز به تعامل است. اثبات صفر دانش کمک می‌کند تا یک نفر به بقیه افراد درست بودن یک جمله را اثبات کند بدون آن که اطلاعات محرمانه‌ای از آن جمله فاش شود. برای مثال اگر افرادی دنبال عددی باشند که چکیده آن برابر یک مقدار مشخص باشد، هر فردی که از این عدد اطلاع داشته باشد

[1] ledger [2] distributed ledger [3] miner [4] hash [5] Zerocash [6] Zero-Knowledge Succinct Non-interactive Argument of Knowledge (ZK-SNARK) [7] note [8] commitment

[9] nonce [10] trapdoor [11] nullifier [12] JoinSplit

می‌تواند با استفاده از زی‌کی-اسنارک بدون فاش کردن آن به افراد مورد نظر اثبات کند که عدد متناظر با آن چکیده را می‌داند. در این الگوریتم بدون دانستن مدرک نمی‌توان اثباتی را ساخت که بعدا توسط سایر افراد تایید شود.

الگوریتم زی‌کی-اسنارک دارای سه مرحله اصلی می‌باشد [۱۵]:

**تولید کلید:** در این مرحله دو کلید مدرک و اعتبارسنجی ساخته می‌شود و به صورت عمومی منتشر می‌شود. برای ساخت این دو کلید دو ورودی مورد نیاز است، یک برنامه و یک پارامتر مخفی برای آن برنامه.

**تولید مدرک:** این مرحله توسط فردی که می‌خواهد ادعای خود را برای دیگران ثابت کند انجام می‌شود و با استفاده از کلید مدرک، پارامتر عمومی و پارامتر خصوصی مدرکی را تولید می‌کند.

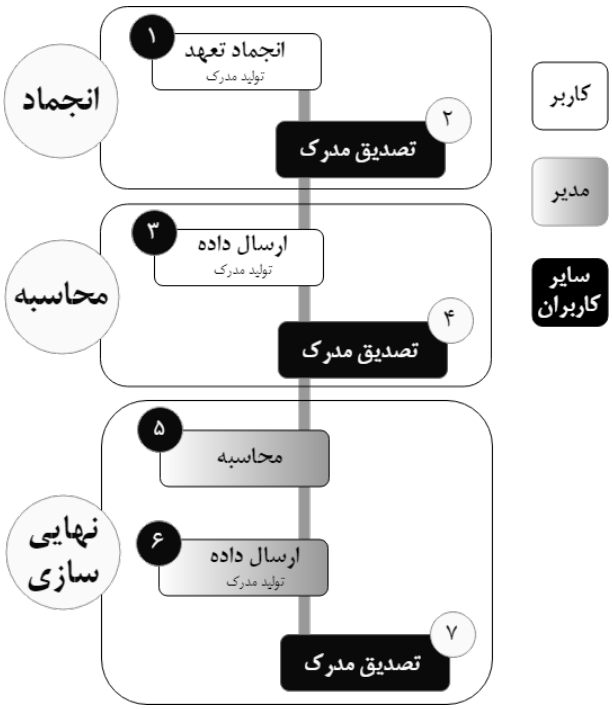
**اعتبارسنجی:** سایر افرادی که می‌خواهند اثبات تولید شده در مرحله قبل را اعتبارسنجی کنند و ادعای فرد مورد نظر را تایید نمایند با استفاده از اثبات تولید شده در مرحله قبل، کلید اعتبارسنجی و پارامتر عمومی شروع به اعتبارسنجی می‌کنند.

زی‌کی-اسنارک دارای قابلیت اعتبارسنجی عمومی است بدین معنا که هر کسی بدون نیاز به تعامل با فرد ادعاکننده می‌تواند ادعای آن را اعتبارسنجی کند. برای تولید مدرک در زی‌کی-اسنارک ابتدا باید مدارهای حسابی [1] برنامه مورد نظر تولید شوند. منظور از مدارهای حسابی، تبدیل گام‌های منطقی توابع مورد نظر به کوچکترین عملیات‌های ممکن می‌باشد. این عملیات‌های ریاضی شامل جمع، تفریق، ضرب و تقسیم می‌باشند. مرحله بعد ساخت ثابت‌هایی هستند که به R1CS [2] معروف می‌باشند. از طریق این ثابت‌ها چندجمله‌ای‌هایی ساخته می‌شود که در تولید کلیدهای اثبات و اعتبارسنجی نقش مهمی دارند و به QAP [3] معروف هستند [۱۶]. برای انجام مراحل بعد از تولید مدارهای حسابی می‌توان از کتابخانه لیب‌سنارک [4] استفاده کرد و زی‌کَش نیز از این کتابخانه برای انجام عملیات‌های خود استفاده نموده است.

### ۴.۲ هاک

مدل هاک متناسب با ساختار رمزارز زی‌کَش طراحی شده است و می‌تواند هر قرارداد هوشمندی را اجرا کند. در این حالت تراکنش‌های مالی به همراه داده‌های قرارداد هوشمند دارای محرمانگی و حریم خصوصی می‌باشند. بدین ترتیب کاربران به همراه ناشناس بودن می‌توانند سطح بالایی از امنیت و حریم خصوصی را در سیستم بانکی غیرمتمرکز تجربه کنند.

در این پروتکل سه جزء اساسی وجود دارد: ۱) کاربران، که شامل تمامی گره‌های شرکت‌کننده در قرارداد هوشمند و افراد تصدیق‌کننده نتایج قرارداد هوشمند می‌باشد، ۲) زنجیره بلوکی، که توسط تمام گره‌های اجماع اجرا می‌شود و تمامی تراکنش‌ها و نتایج قرارداد هوشمند در آن ثبت




**شکل ۱.** فرایند پروتکل هاک در حفظ حریم خصوصی قراردادهای هوشمند

می‌شود و ۳) جزء مدیر، که برای اجرای قرارداد هوشمند به کار می‌رود. هر قرارداد هوشمند مدیر خودش را دارد و هر مدیر می‌تواند همزمان فقط یک قرارداد هوشمند را اجرا کند. همه کاربران موجود در شبکه می‌توانند به عنوان مدیر فعالیت کنند. قراردادهای هوشمند نیز از دو بخش جداگانه تشکیل شده‌اند: بخش خصوصی که در آن محاسباتی که باید دارای حریم خصوصی باشند قرار گرفته است و بخش عمومی که جریمه‌ها و طریقه برخورد با متخلفان را مشخص می‌کند. قسمت خصوصی قرارداد هوشمند توسط مدیر اجرا می‌شود ولی قسمت عمومی توسط معدن‌کاو که وظیفه ساخت بلوک بعدی را برعهده دارد اجرا می‌شود. پروتکل هاک دارای سه مرحله اساسی می‌باشد که در شکل ۱ نشان داده شده است [۱۲]. در ادامه به توضیح هر یک از این مراحل می‌پردازیم.

#### ۱.۴.۲ انجماد

در این مرحله یک کاربر ارز مورد نظر خود را از استخر خصوصی برداشته و آن را به لیستی به نام سکه‌های منجمدشده اضافه می‌کند. این مقدار از پول برای ورودی قرارداد هوشمند استفاده می‌شود. همچنین هر کاربر می‌تواند فقط یک ورودی به قرارداد هوشمند ارسال کند. ارسال ورودی باید همراه با تولید اثبات برای درستی مقدار مورد نظر باشد. تصدیق این اثبات باید از طرف اکثریت اعضای شبکه صورت گیرد تا مرحله بعد پروتکل اجرا شود. این تصدیق در شروع دوره زمانی جدید اتفاق می‌افتد. در زنجیره بلوکی هر دوره زمانی متعلق به ساخت و افزودن بلوک جدید به زنجیره بلوکی می‌باشد.

[1] arithmetic circuit [2] Rank One Constraint System [3] Quadratic Arithmetic Programs [4] Libsnark

### ۲.۴.۲ محاسبه

در این مرحله ورودی منجمد شده کاربر به مدیر قرارداد هوشمند مورد نظر ارسال می‌شود. این مقادیر با کلید عمومی مدیر رمز می‌شود تا دیگر اعضا در زنجیره بلوکی از محتوای آن‌ها اطلاعی نداشته باشند. در این مرحله نیز کاربر باید برای اثبات ارسال صحیح مقادیر به مدیر مدرکی تولید و آن را برای اعتبارسنجی به کاربران ارسال کند. در صورت تصدیق این اثبات ورودی کاربر توسط مدیر مورد قبول واقع می‌شود. این تصدیق نیز در شروع دوره زمانی جدید انجام می‌شود.

### ۳.۴.۲ نهایی‌سازی

در این مرحله مدیر پس از دریافت تمام ورودی‌ها و رمزگشایی آن‌ها شروع به انجام محاسبات و بدست آوردن خروجی و نتیجه قرارداد هوشمند می‌کند. سپس آن‌ها را برای هر یک از کاربران به صورت رمز شده ارسال می‌کند و نتایج در زنجیره بلوکی خصوصی ذخیره می‌شود. مدیر فقط قسمت خصوصی قرارداد هوشمند را اجرا می‌کند که با داده‌های حساس و محرمانه کاربران کار می‌کند. در نهایت برای اثبات درستی اجرای قرارداد هوشمند، مدرکی را تولید و به کاربران ارسال می‌کند. در صورت عدم تصدیق توسط اکثریت، مدیر متخلف شناخته شده و باید جریمه‌ای به طرفین قرارداد پرداخت کند. همچنین بدلیل اطلاع مدیر از داده‌های ورودی، نتایج و داده‌های خروجی، افشای اطلاعات کاربران جزئی از تخلف محسوب می‌شود که باعث نقض حریم خصوصی کاربران می‌شود. البته این افشای اطلاعات هیچ تاثیری در نتیجه اجرای قرارداد هوشمند ندارد و کاربران می‌توانند داده‌ها و نتایج دریافتی از مدیر را تغییر دهند تا از این سوء استفاده در امان باشند.

## ۳ راه حل پیشنهادی

یکی از اصلی‌ترین مشکلات موجود در مدل هاک عدم تطابق با نسخه بروز و تغییریافته زی‌کش می‌باشد به طوری که پیاده‌سازی آن را با مشکلاتی همانند عدم توانایی جزء مدیر در تولید نتایج قرارداد هوشمند مواجه کرده است. همچنین نحوه تعیین جزء مدیر و به اجماع رسیدن بر روی یک مدیر مشخص برای اجرای قرارداد هوشمند شفاف نمی‌باشد و آسیب‌پذیری‌های مختلفی همچون حمله مرد میانی[1] در این بین به وجود می‌آید. علاوه بر این، اجرای قرارداد هوشمند برای مدیر سربار محاسباتی بسیار بالایی دارد.

در این بخش از نوشتار، پروتکل پیشنهادی برای حفظ حریم خصوصی در قراردادهای هوشمند بر پایه زی‌کش و هاک ارائه می‌شود. تغییرات انجام شده باعث رفع مشکلات ذکرشده و کاهش سربار محاسباتی اجرای قرارداد هوشمند در سمت مدیر می‌شود. تمام شبه‌کدهای ارائه‌شده در این قسمت که دارای پس‌زمینه خاکستری می‌باشند در شروع دوره زمانی جدید اجرا می‌شوند. همچنین این قسمت از شبه‌کدها می‌تواند توسط هر گره‌ای در زنجیره بلوکی اجرا شود و کاربران باید برای تکمیل فرایند اجرای قرارداد هوشمند به اجماع کلی برسند. بنابراین در پایان هر دوره زمانی عملیات‌های قسمت مورد نظر باید با موفقیت تمام شده باشند. در غیر این صورت کاربر مورد نظر نمی‌تواند در قرارداد هوشمند مد نظرش شرکت کند. در مرحله اول جزء مدیر جفت کلید رمزنگاری نامتقارن خود را تولید نموده و آدرس و کلید عمومی تولید شده را برای تمام کاربران در شبکه ارسال می‌کند. انتخاب جزء مدیر توسط صاحب قرارداد هوشمند انجام می‌شود و آدرس آن در داخل قرارداد هوشمند ثبت می‌شود. هر گره‌ای پس از دریافت اطلاعات گره مدیر آن را به گره‌های جفت خود ارسال می‌کند.

در مرحله بعد هر کاربری که می‌خواهد در مزایده شرکت کند باید پیشنهاد خود را به اصطلاح منجمد نموده و آن را به لیست پول‌های خرج شده اضافه نماید که به این مرحله انجماد گفته می‌شود. این قسمت از الگوریتم دارای چند زیرمرحله می‌باشد:

(۱) ایجاد یادداشت زیر با مقدار پیشنهادی مورد نظر برای انجماد

$$Note(a_{pk}, value, rho, s, JsoutPoint)$$

که در آن $JsoutPoint$ ساختاری می‌باشد که دارای اطلاعات مختصری از تراکنش مولد یادداشت مورد نظر است. مقادیر این ساختار در تولید نتایج توسط جزء مدیر مورد استفاده قرار می‌گیرد.
(۲) تولید مقادیر تعهد و ابطال‌کننده مطابق با رابطه‌های (۱) و (۳)،
(۳) تولید جفت کلید عمومی و خصوصی جوین‌اسپلیت برای انجام تراکنش‌های نهایی تولیدی توسط مدیر،
(۴) ذخیره اطلاعات مربوط به یادداشت، کلیدهای جوین‌اسپلیت و تعهد مورد نظر،
(۵) تولید ورودی‌های اولیه و کمکی برای زی‌کی-اسنارک،
(۶) تولید مدرک با استفاده از ورودی‌های اولیه و کمکی،
(۷) انتشار مدرک تولید شده، تعهد و ابطال‌کننده مورد نظر،
(۸) بررسی اولین انجماد برای کاربر مورد نظر توسط سایر گره‌ها،
(۹) اعتبارسنجی مدرک توسط تمام گره‌ها و رسیدن به اجماع،
(۱۰) و افزودن تعهد و ابطال‌کننده مورد نظر به لیست انجماد.

برای مرحله انجماد کافی است مدرکی تولید شود که نشان دهد ابطال‌کننده تولید شده مناسب تعهد اعلام شده می‌باشد. همچنین باید مسیر درخت مرکل از ریشه تا تعهد مورد نظر بررسی شده و در مورد درستی آن به اجماع رسید. شایان ذکر است صحت تعهد تولید شده قبلا در تراکنش جوین‌اسپلیت بررسی شده و نیازی به تولید مدرک برای اثبات درستی آن نمی‌باشد.

برای تولید مدرک نیاز به تولید دو ورودی اولیه و کمکی می‌باشد. این دو گروه ورودی برای اثبات‌کننده آشکار می‌باشند اما ورودی کمکی برای کاربرانی که عملیات اعتبارسنجی را انجام می‌دهند محرمانه می‌ماند. در مرحله انجماد ورودی‌های اولیه و کمکی به صورت زیر تعیین می‌شوند:

**ورودی‌های اولیه:** ریشه درخت مرکل، تعهد منجمد شده و ابطال‌کننده متناظر،

**ورودی‌های کمکی:** کلید خصوصی، کلید عمومی، مقدار پیشنهادی،

[1] Man in the Middle (MITM)

شاخه درخت مرکل مربوط به تعهد مورد نظر و تک‌شمار تصادفی یادداشت مورد نظر.

در مرحله بعد کاربری که پیشنهاد خود را منجمد نموده و توسط سایر کاربران راستی‌آزمایی شده است شروع به ارسال داده به جزء مدیر می‌کند که به اصطلاح به این مرحله محاسبه کاربر گفته می‌شود. این مرحله نیز دارای چند گام به شرح زیر می‌باشد:

(۱) رمزگذاری تمام داده‌های مورد نیاز مدیر با کلید عمومی منتشر شده (مقدار پیشنهاد، اطلاعات یادداشت، آدرس خصوصی، کلید عمومی جوین‌اسپلیت و کلید رمزنگاری متقارن AES)،
(۲) تولید ورودی‌های اولیه و کمکی برای زی‌کی-اسنارک،
(۳) تولید مدرک با استفاده از ورودی‌های اولیه و کمکی،
(۴) انتشار مدرک تولید شده و متن رمز شده،
(۵) اعتبارسنجی مدرک توسط تمام گره‌ها و رسیدن به اجماع،
(۶) و ارسال متن رمز شده به جزء مدیر.

در این مرحله مدرک تولید شده فقط برای اثبات درستی مقادیر رمز شده می‌باشد. این مقادیر باید متناظر با مقادیر منجمد شده در مرحله انجماد باشد. همچنین ورودی‌های اولیه و کمکی به صورت زیر تعیین می‌شوند:

**ورودی‌های اولیه:** متن رمز شده و تعهد منجمد شده،
**ورودی‌های کمکی:** کلید عمومی، مقدار پیشنهادی و تک‌شمار تصادفی یادداشت.

تا پایان مرحله دوم، دو دوره زمانی طی می‌شود و دو بلوک به زنجیره بلوکی اضافه می‌شود. در مرحله سوم جزء مدیر شروع به رمزگشایی متن‌های رمز شده کرده و محاسبات مورد نظر را انجام می‌دهد. این مرحله نیز شامل چند زیرمرحله به صورت زیر می‌باشد:

(۱) رمزگشایی متن‌های رمز شده و ذخیره داده‌ها،
(۲) اجرای قرارداد هوشمند مزایده دومین قیمت بر اساس پیشنهادهای ارائه شده و انتخاب بیشترین پیشنهاد و محاسبه آن با دومین پیشنهاد بالا،
(۳) محاسبه و ارسال مقادیر تراکنش‌های حاصل از اجرای قرارداد هوشمند:
  (آ) مقدار تراکنش ارسالی به برنده مزایده = تفاضل مقدار پیشنهادی و مقدار دومین پیشنهاد،
  (ب) مقدار تراکنش ارسالی به حساب صاحب قرارداد هوشمند = مقدار دومین پیشنهاد،
  (ج) مقدار تراکنش ارسالی به سایر طرف‌های شرکت کننده در مزایده = مقدار پیشنهادی خودشان (می‌توان مقداری را به عنوان هزینه شرکت در مزایده از طرفین دریافت کرد و به جزء مدیر داد.)،
(۴) تولید مولفه‌های مورد نیاز برای تراکنش جوین‌اسپلیت و محاسبه تعهدات متناظر،
(۵) رمزگذاری مولفه‌های تراکنش جوین‌اسپلیت با رمز متقارن AES و کلید متقارن ارسالی هر کاربر،
(۶) تولید مدرک با استفاده از ورودی‌های اولیه و کمکی،
(۷) اعلام برنده مزایده و انتشار اثبات صحت اجرای قرارداد و تعهدهای تولید شده جدید به صورت آشکار همراه با مولفه‌های رمز شده،
(۸) اعتبارسنجی مدرک توسط تمام گره‌ها و رسیدن به اجماع،
(۹) و انجام تراکنش‌ها توسط هر گره با توجه به مولفه‌های دریافتی.

در این مرحله (مرحله سوم)، مدیر کافی است مدرکی برای اثبات درستی اجرای قرارداد هوشمند تولید نماید و از آن طریق اثبات کند که مقادیر تولیدی برای طرفین قرارداد درست می‌باشد. برای تولید این مدرک می‌توان از سیستم اعتبارسنجی پپر پیکوین [1] استفاده نمود [۱۷]. در این سیستم قرارداد هوشمند مزایده دومین قیمت را با زبان C++ پیاده‌سازی نموده و متناسب با ورودی‌ها و خروجی‌ها مدرک مورد نظر تولید می‌شود. تمامی اعمال زی‌کی-اسنارک از جمله تولید مدارات حسابی، تولید بردارهای R1CS، تولید چندجمله‌ای‌ها و تولید کلیدهای مورد نظر به صورت خودکار و با سرعت بالایی انجام می‌شود. این سیستم از کامپایلر پینوکیو [2] برای تولید مدارات حسابی برنامه مورد نظر استفاده می‌کند. برای افزایش سرعت تولید مدارات حسابی یک نسخه بهینه از این کامپایلر در این سیستم پیاده‌سازی شده است [۱۸].

سیستم پیکوین دارای دو نوع ورودی عمومی و خصوصی می‌باشد که باید توسط تولیدکننده مدرک (مدیر) تکمیل شود. مدیر مقادیر پیشنهادی کاربران را در ورودی خصوصی پیکوین قرار می‌دهد. این ورودی فقط توسط تولیدکننده مدرک (مدیر) قابل مشاهده می‌باشد و اعتبارسنج‌ها از مقدار آن مطلع نمی‌باشند. ورودی عمومی نیز توسط مدیر تکمیل می‌شود با این تفاوت که مقدار آن برای همگان آشکار است و در فرایند اعتبارسنجی نیز استفاده می‌شود. برای جلوگیری از تقلب و اجرای درست قرارداد هوشمند مزایده دومین قیمت با ورودی‌های صحیح، مقدار ورودی عمومی سیستم تولید مدرک به صورت رابطه (۴) محاسبه شده و توسط مدیر تکمیل می‌شود.

$$public\ input_i := \mathrm{SHA256}(value_i, s_i) \qquad (۴)$$

که در این رابطه $value_i$ مقدار پیشنهادی کاربر $i$ام، $s_i$ تک‌شمار تصادفی تعهد کاربر $i$ام و $public\ input_i$ سطر $i$ام از ورودی عمومی را نشان می‌دهد. همچنین در برنامه مزایده دومین قیمت قبل از محاسبه برنده مزایده، مقادیر خصوصی را به صورت رابطه (۴) محاسبه نموده و نتیجه را با مقادیر ورودی عمومی مقایسه می‌کنیم. در صورتی اجرای مزایده انجام می‌پذیرد که تمامی مقایسه‌های مورد نظر درست باشند و مقادیر پیشنهادی کاربران به صورت صحیح، توسط مدیر در ورودی خصوصی وارد شده باشند. بدین ترتیب مقدار $s_i$ نیز باید در ورودی خصوصی وجود داشته باشد. با این تکنیک هر کاربری که در قرارداد هوشمند مورد نظر شرکت کرده باشد می‌تواند با بررسی صحیح بودن ورودی عمومی از اجرای صحیح

[1] Pepper Pequin [2] Pinocchio Compiler

قرارداد هوشمند اطمینان حاصل کند. خروجی این سیستم نیز باید نتایج حاصل از اجرای قرارداد هوشمند باشد. به منظور جلوگیری از افشای این اطلاعات، نتایج به صورت رابطه (۵) محاسبه شده و در خروجی قرار می‌گیرد.

$$output_i := \text{SHA256}(newvalue_i, s_i) \qquad (۵)$$

برای اعتبارسنجی درستی انجام تراکنش جوین‌اسپلیت توسط هر کاربر با توجه به مولفه‌های تولیدی مدیر، کاربران می‌توانند پس از انجام تراکنش جوین‌اسپلیت تعهد تولیدی توسط کاربران را با تعهد تولید شده توسط مدیر مقایسه نموده و به درستی عملیات پی ببرند. در پایان، کاربران برای حفظ امنیت و حریم خصوصی خود در برابر جزء مدیر می‌توانند یادداشت‌های جدیدی با مولفه‌های جدید تولید کرده تا از افشای اطلاعات حساس توسط مدیر و سوء استفاده توسط دیگران جلوگیری به عمل آورند. با این حال افشای مولفه‌های تولیدی توسط مدیر هیچ آسیب امنیتی به شبکه وارد نمی‌کند. دلیل این مسئله عدم اطلاع دیگران از کلید خصوصی کاربر و عدم توانایی در تولید ابطال‌کننده بدون کلید خصوصی می‌باشد.

یکی از مهم‌ترین مزیت‌های رمزارزها، غیرمتمرکز بودن آن‌ها می‌باشد به طوری که هر یک از کاربران می‌تواند سهمی در مدیریت و ثبت تراکنش‌ها داشته باشد. در مدل پیشنهادی برای حفظ حریم خصوصی در قراردادهای هوشمند وجود جزء مدیر برای اجرای قرارداد هوشمند و تولید نتایج تا حد زیادی از غیرمتمرکز بودن شبکه کم می‌کند. با این حال به دلیل مشارکت سایر کاربران در تمام مراحل اجرای قرارداد هوشمند به صورت اعتبارسنج، غیرمتمرکز بودن شبکه تا حدودی حفظ شده و سایر کاربران می‌توانند سهمی در مدیریت اجرای قرارداد هوشمند داشته باشند.

## ۴ نتایج ارزیابی

علاوه بر سربار تراکنش‌های زی‌کَش، زمان اجرای قرارداد هوشمند به سه عامل دیگر وابسته می‌باشد. این سه عامل تولید مدرک در سه مرحله مختلف اجرای قرارداد هوشمند می‌باشد. هزینه زمانی اعتبارسنجی مدارک تولید شده بسیار کم و قابل چشم‌پوشی است اما برای تولید هر یک از مدارک با استفاده از زی‌کِی–اسنارک زمان زیادی صرف می‌شود. استفاده از ابزارهای کاراتر برای تولید مدارک می‌تواند کمک شایانی به کاهش سربار تولید اثبات کند. همانطور که در قسمت قبل گفته شد ما برای تولید مدرک در سمت مدیر از سیستم پیکوین استفاده کرده‌ایم که باعث کاهش ۴۷ درصدی زمان اجرای قرارداد هوشمند شده است. در مقابل، در سمت کاربران تغییری انجام نشده است و از همان سیستم زی‌کِی–اسنارک زی‌کَش برای تولید مدرک استفاده نموده‌ایم. جدول ۱ سربار زمانی اجرای قرارداد هوشمند مزایده دومین قیمت در سمت مدیر را به ازای ۱۰ شرکت‌کننده نشان می‌دهد. این زمان اجرا با مدل اصلی هاک مقایسه شده است.

**جدول ۱.** سربار زمانی اجرای قرارداد هوشمند مزایده دومین قیمت برای ۱۰ شرکت‌کننده در سمت مدیر

| | مدل پیشنهادی | مدل هاک | بهبود (٪) |
|---|---|---|---|
| تولید کلید (ثانیه) | ۱۷٫۳ | ۳۲٫۳ | ۴۶ |
| تولید مدرک (ثانیه) | ۸٫۲ | ۱۵٫۴ | ۴۷ |
| اعتبارسنجی (میلی‌ثانیه) | ۱۴ | ۱۰ | −۴۰ |
| مجموع (ثانیه) | ۲۵٫۵۱ | ۴۷٫۷۱ | ۴۷ |

## ۵ نتیجه‌گیری و کارهای آتی

اجرای صحیح بعضی از قراردادهای هوشمند نیازمند حفظ حریم خصوصی طرفین قرارداد می‌باشد. بخشی از این حریم خصوصی توسط تراکنش‌های حفاظت شده در زی‌کَش تامین شده است. برای اعمال کامل حریم خصوصی می‌توان از مدل هاک استفاده نمود. در این مقاله نسخه‌ای قابل پیاده‌سازی و شخصی‌سازی شده از این مدل برای قرارداد هوشمند مزایده دومین قیمت ارائه شد که می‌توان آن را بر روی رمزارز زی‌کَش پیاده‌سازی کرد. این نسخه علاوه بر کاهش پیچیدگی‌ها در پیاده‌سازی، موجب بهبود نزدیک به ۵۰ درصدی در زمان اجرای قرارداد هوشمند در سمت مدیر می‌شود.

راهکارهای دیگری برای بهبود زمان اجرا و کم کردن سربار زی‌کِی–اسنارک وجود دارد که می‌توان برای ادامه مسیر از آن‌ها استفاده نمود. یکی از این راهکارها استفاده از زی‌کِی–استارک [1] [۱۹] می‌باشد. این روش اثبات صفر دانش برای استفاده در زنجیره‌های بلوکی بسیار مناسب است و در تولید اثبات ده برابر سریع‌تر از زی‌کِی–اسنارک عمل می‌کند. همچنین در زی‌کی–استارک مرحله راه‌اندازی به طور کلی حذف شده است و دیگر سربار زمانی برای تولید کلیدهای مدرک و اعتبارسنجی وجود ندارد [۱۹]. با این حال پیاده سازی این نوع اثبات صفر دانش بسیار مشکل می‌باشد.

برای افزایش غیرمتمرکز بودن مدل پیشنهادی می‌توان از محاسبات چندجانبه امن در سمت مدیر استفاده کرد [۲۰]. همچنین استفاده از بسترهای امن سخت‌افزاری مانند اینتل SGX برای افزایش امنیت در سمت مدیر مفید می‌باشد.

## مراجع


[1] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *Ieee Access*, 4:2292–2303, 2016.

[2] Harry Halpin and Marta Piekarska. Introduction to security and privacy on the blockchain. In *2017 IEEE European Symposium on Security and Privacy Workshops*


[1] ZK-STARK

[13] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. Blockchain for iot security and privacy: The case study of a smart home. In *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, pages 618–623. IEEE, 2017.

[14] D Hopwood, S Bowe, T Hornby, and N Wilcox. Zcash protocol specification (version 2018.0-beta-9). 2018.

[15] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. Snarks for c: Verifying program executions succinctly and in zero knowledge. In *Annual cryptology conference*, pages 90–108. Springer, 2013.

[16] Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure sampling of public parameters for succinct zero knowledge proofs. In *2015 IEEE Symposium on Security and Privacy*, pages 287–304. IEEE, 2015.

[17] M Howald and F Leupold. Pequin: an end-to-end toolchain for verifiable computation, snarks, and probabilistic proofs. `https://github.com/pepper-project/pequin`, 2016.

[18] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. *Communications of the ACM*, 59(2):103–112, 2016.

[19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. *Cryptology ePrint Archive*, 2018.

[20] Ben Kreuter, Abhi Shelat, Benjamin Mood, and Kevin Butler. {PCF}: A portable circuit format for scalable {Two-Party} secure computation. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 321–336, 2013.

*(EuroS&PW)*, pages 1–3. IEEE, 2017.

[3] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[4] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE symposium on security and privacy*, pages 459–474. IEEE, 2014.

[5] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1, 2014.

[6] Guy Zyskind, Oz Nathan, et al. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184. IEEE, 2015.

[7] Hisham S Galal and Amr M Youssef. Verifiable sealed-bid auction on the ethereum blockchain. In *International Conference on Financial Cryptography and Data Security*, pages 265–278. Springer, 2018.

[8] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In *International conference on financial cryptography and data security*, pages 79–94. Springer, 2016.

[9] Guy Zyskind, Oz Nathan, and Alex Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471*, 2015.

[10] Rui Yuan, Yu-Bin Xia, Hai-Bo Chen, Bin-Yu Zang, and Jan Xie. Shadoweth: Private smart contract on public blockchain. *Journal of Computer Science and Technology*, 33(3):542–556, 2018.

[11] Raymond Cheng, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah Johnson, Ari Juels, Andrew Miller, and Dawn Song. Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 185–200. IEEE, 2019.

[12] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE symposium on security and privacy (SP)*, pages 839–858. IEEE, 2016.